



TUGAS AKHIR -SM141501

**SUPER RESOLUSI OBYEK BERBASIS CITRA TRACKING
MENGUNAKAN METODE *PHASED BASED IMAGE
MATCHING* DAN METODE PROYEKSI PADA HIMPUNAN
CONVEX**

HASANUDDIN AL-HABIB
NRP 1211 100 083

DosenPembimbing
Dr. Budi Setiyono, S.Si, MT

JURUSAN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2015



FINAL PROJECT -SM141501

***SUPER RESOLUTION IMAGE BASED OBJECT TRACKING
USING PHASED BASED IMAGE MATCHING METHOD AND
PROJECTION ONTO CONVEX SETS METHOD***

HASANUDDIN AL-HABIB
NRP 1211 100 083

Supervisor
Dr. Budi Setiyono, S.Si, MT

DEPARTMENT OF MATHEMATICS
Faculty of Mathematics and Natural Science
Sepuluh Nopember Institute of Technology
Surabaya 2015

LEMBAR PENGESAHAN
SUPER RESOLUSI OBYEK BERBASIS CITRA
TRACKING MENGGUNAKAN METODE
PHASED BASED IMAGE MATCHING DAN
METODE PROYEKSI PADA HIMPUNAN
CONVEX
SUPER RESOLUTION IMAGE BASED OBJECT
TRACKING USING PHASED BASED IMAGE
MATCHING METHOD AND PROJECTION ONTO
CONVEX SETS METHOD

Diajukan Untuk Memenuhi Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Sains
Pada Bidang Studi Matematika Terapan
Program Studi S-1 Jurusan Matematika
Fakultas Matematika Dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember Surabaya

Oleh:
HASANUDDIN AL-HABIB
NRP. 1211 100 083

Menyetujui,

Pembimbing

Dr. Budi Setiyono, S.Si., M.T.
NIP. 19720207 199702 1 001

Mengetahui,

Kepala Jurusan Matematika
MIPA ITS

Prof. Dr. Irena Apriliani, M.Si

JURUSAN. 19660414 199102 2 001

MATEMATIKA

Surabaya, Juli 2015

SUPER RESOLUSI OBYEK BERBASIS CITRA TRACKING MENGGUNAKAN METODE PHASED BASED IMAGE MATCHING DAN METODE PROYEKSI PADA HIMPUNAN CONVEX

Nama Mahasiswa : Hasanuddin Al-Habib
NRP : 1211 100 083
Jurusan : Matematika
Dosen Pembimbing : Dr. Budi Setiyono, S.Si, MT

Abstrak

Super resolusi merupakan teknik untuk mendapatkan citra beresolusi tinggi dari citra yang beresolusi rendah. Citra resolusi rendah yang digunakan dapat berupa citra tunggal maupun rangkaian citra yang diambil dari *scene* yang sama agar citra tersebut menyediakan informasi yang sama untuk proses rekontruksi citra resolusi tinggi. Pada umumnya super resolusi dilakukan pada keseluruhan piksel *frame*, padahal kebutuhan informasi dari suatu citra hanya terdapat pada suatu bagian tertentu dari *frame* tersebut bukan pada keseluruhan piksel *frame*. Oleh karena itu, agar proses super resolusi dapat lebih efektif maka digunakan ROI (*Region Of Interest*) untuk mendapatkan bagian dari *frame* dan melakukan proses *tracking* dalam rangkaian citra. Pada tugas akhir ini digunakan metode *phased based image matching* untuk proses registrasi dan metode *projection onto convex sets* untuk proses rekonstruksi. Pada pengujian yang dilakukan nilai rata-rata PSNR diperoleh 30,3183 dB dan nilai PSNR akan semakin besar jika citra observasi semakin banyak. Selain itu, diperoleh waktu komputasi citra *tracking* sebesar 7,456 detik sedangkan pada keseluruhan piksel pada frame citra sebesar 188,306 detik. Hal ini menunjukkan bahwa waktu komputasi citra *tracking* lebih efektif dibandingkan keseluruhan piksel pada frame citra

Kata Kunci—Objek *tracking*, Super resolusi, *normalized cross correlation*, *phased based image matching*, *projection onto convex sets*, ROI.

“Halaman ini sengaja dikosongkan”

SUPER RESOLUTION IMAGE BASED OBJECT TRACKING USING PHASED BASED IMAGE MATCHING METHOD AND PROJECTION ONTO CONVEX SETS METHOD

Name : Hasanuddin AL-Habib
NRP : 1211 100 083
Department : Mathematics
Supervisor : Dr. Budi Setiyono, S.Si, MT

Abstract

Super-resolution is a technique to obtain high-resolution images from low-resolution image. Low-resolution images used may be a single image or series of images taken of the same scene so that the image provides the same information for a high resolution image reconstruction process. In general, carried out on the overall super-resolution pixel frame, whereas the information needs of an image only in a certain part of the frame rather than the whole pixel frame. Therefore, in order to super-resolution process can be more effective then used ROI (Region Of Interest) to get a piece of the frame and make the process of tracking in a series of images. In this final project, its implemented phased based image matching method for the registration process and the projection onto convex sets method for the reconstruction process. In the tests performed, we got the average of PSNR value is 30,3183 dB and this PSNR value will increase if the number of observations image increase. In addition, images obtained by computing time tracking of 7.456 second while the total pixels in the image frame at 188.306 second. This indicates that the computing time tracking image is more effective than a whole pixel in the image frame.

Keywords: Tracking objek, Super-resolution, normalized cross correlation, phased based image matching, projection onto convex sets, ROI.

“Halaman ini sengaja dikosongkan”

KATA PENGANTAR

Segala Puji bagi Allah SWT yang telah memberikan karunia, rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul: **“Super Resolusi Obyek Berbasis Citra Tracking Menggunakan Metode *Phased Based Image Matching* Dan Metode Proyeksi Pada Himpunan Convex”** yang merupakan salah satu persyaratan akademis dalam menyelesaikan Program Studi S-1 pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Institut Teknologi Sepuluh Nopember Surabaya.

Tugas Akhir ini dapat diselesaikan dengan berkat kerjasama, bantuan, dan dukungan dari banyak pihak. Sehubungan dengan hal itu, penulis mengucapkan terima kasih kepada:

1. Dr. Budi Setiyono, S.Si, MT selaku dosen pembimbing yang senantiasa membimbing dengan sabar dan memberikan kritik dan saran dalam penyusunan Tugas Akhir ini.
2. Prof. Dr. Erna Apriliani, M.Si selaku Ketua Jurusan Matematika.
3. Drs. Sadjidon, M. Si selaku Dosen Wali.
4. Prof. DR. Isa Irawan, S.Si, M.T, Drs. Sadjidon, M. Si, Drs. Sentot Didik Surjanto, M.Si dan Dr. Dwi Ratna Sulistyaningrum, S.Si, MT selaku dosen penguji Tugas Akhir ini.
5. Dr. Chairul Imron, MI.Komp. selaku Koordinator Tugas Akhir.
6. Seluruh jajaran dosen dan staf jurusan Matematika ITS.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan saran dan kritik dari pembaca. Akhir kata, semoga Tugas Akhir ini bermanfaat bagi semua pihak yang berkepentingan.

Surabaya, Juli 2015

Penulis

special thanks to

Selama proses pembuatan Tugas Akhir ini, banyak pihak yang telah memberikan bantuan dan dukungan untuk penulis. Penulis mengucapkan terima kasih dan apresiasi secara khusus kepada:

1. Ayah Mohamad Subandi dan Ibu Ngarti tersayang yang senantiasa dengan ikhlas memberikan semangat, perhatian, kasih sayang, doa, motivasi dan nasihat yang sangat berarti bagi penulis.
2. Nenek Reni, Adik Abdul Aziz Alhaqi dan Zahrotul Lailin Nadhiroh yang menjadi salah satu motivasi bagi penulis untuk segera menyelesaikan Tugas Akhir ini.
3. Eni mariana yang telah meluangkan waktu untuk mengingatkan penulis untuk bimbingan dan mengerjakan tiap *step* tugas akhir ini.
4. Liyana Wulandhari, Yahya Efprianto, Wiwid Setyawan, Satria Stanza Pramayoga yang sudah membantu dalam mengerjakan program dan hitungan manual dalam tugas akhir ini.
5. Teman-teman kontrakan Kejawan Gebang No. 20: Singgih, Hakam, Isman, Ilham, Anas, Jamil, Heri, Rifdy, Zamroji. Semoga sukses kedepannya dan selalu dalam perlindungan Allah swt.
6. Kabinet Sahabat Himatika ITS 13/14 : Isman Safii, Yahya Efprianto, Ilham Fauzi H., Mukhofifatul M., Nur Dina Mustifa, Ahmad Farid M., Heri Purnawan, Liyana Wulandhari, Zain Rizqiyyah, Azaria Natasha, Lena Desy A., Auliya Rahmayani. Terima kasih atas kekeluargaan dan pengalaman berharga dalam satu kepengurusan.
7. Fendi, Dona, Vimala, Mardiana, Handy, Selvi, dan Teman-teman Matematika ITS 2011 serta teman-teman satu dosen pembimbing: Farah tetap semangat ya, Inne, dan Mbak Niswa
8. Teman-teman PSDM Himatika ITS 12/13 dan 13/14 sukses selalu.

Tentu saja masih banyak pihak lain yang turut andil dalam penyelesaian tugas akhir ini yang tidak bisa penulis sebutkan satu persatu. Semoga Allah membalas dengan balasan yang lebih baik bagi semua pihak yang telah membantu penulis. *Amin ya rabbal 'alamin.*

“Halaman ini sengaja dikosongkan”

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
LEMBAR PENGESAHAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
DAFTAR LAMPIRAN	xxiii
 BAB I. PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan	4
1.5 Manfaat	4
1.6 Sistematika Penulisan Tugas Akhir	4
 BAB II. TINJAUAN PUSTAKA	
2.1 Pengertian Citra	8
2.2 Citra Digital	9
2.2.1 Digitalisasi Spasial (<i>Sampling</i>).....	9
2.2.2 Digitalisasi Intensitas.....	9
2.3 Video Digital	10
2.4 Resolusi Citra.....	11
2.5 Citra Berwarna dan Citra Grayscale	11
2.6 Transformasi Fourier Diskrit Dua Dimensi	12
2.7 Objek <i>Tracking</i>	13
2.8 <i>Template Matching</i>	13
2.8.1 Metode Normalized Cross Correlation	14
2.9 <i>Region Of Interest</i> (ROI)	14
2.10 <i>Peak Signal to Noise Ratio</i>	15
2.11 Super Resolusi	15

2.12 Metode <i>Phased Based Image Matching</i> (PBIM)	16
2.13 Metode <i>Projection Onto Convex sets</i> Algorithm (POCS)	17
BAB III. METODOLOGI	
3.1 Objek Penelitian	21
3.2 Peralatan	21
3.3 Tahap Penelitian	21
BAB IV. PERANCANGAN DAN IMPLEMENTASI SISTEM	
4.1 Analisis Sistem	25
4.1.1 Analisis Sistem Perangkat Lunak	25
4.1.2 Analisis Kebutuhan Sistem	30
4.2 Perancangan Sistem	31
4.2.1 Gambaran Sistem	31
4.2.2 Penjelasan Umum Sistem	33
4.2.3 Perancangan Proses Algoritma	33
4.2.3.1 Perancangan Proses <i>Tracking</i>	33
4.2.3.2 Perancangan Proses Registrasi Citra	40
4.2.3.3 Perancangan Proses Rekonstruksi Citra	44
4.2.4 Perancangan Data	47
4.2.4.1 Data Masukan	47
4.2.4.2 Data Proses	47
4.2.4.3 Data Keluaran	48
4.2.5 Perancangan Antar Muka Sistem	49
4.2.5.1 Perancangan Halaman Utama	49
4.3 Implementasi Sistem	51
4.3.1 Implementasi <i>Input</i> Video	51
4.3.2 Implementasi Pemilihan Area ROI	52
4.3.3 Implementasi Proses <i>Tracking</i> dengan <i>2D-Normalized Cross Correlation</i>	53
4.3.4 Implementasi Proses Penginputan Citra Observasi untuk Registrasi Super Resolusi	54
4.3.5 Implementasi Registrasi Citra Menggunakan Metode <i>Phased based</i>	

<i>Image Matching</i> (PBIM)	54
4.3.6 Implementasi Rekonstruksi Citra Menggunakan Metode <i>Projection Onto</i> <i>Convex Sets Algorithm</i> (POCS)	56
BAB V. PENGUJIAN DAN PEMBAHASAN	
5.1 Data Uji Coba	57
5.2 Pengujian Tahap Pemilihan ROI	57
5.3 Pengujian Hasil Super Resolusi	58
5.3.1 Pengujian PSNR	58
5.3.2 Pengujian Waktu Komputasi	61
BAB VI. KESIMPULAN DAN SARAN	
6.1 Kesimpulan	65
6.2 Saran	66
DAFTAR PUSTAKA	67
LAMPIRAN	71

“Halaman ini sengaja dikosongkan”

DAFTAR TABEL

	Halaman
Tabel 4.1 Tabel Kebutuhan Sistem	31
Tabel 4.2 Tabel Data Proses	48
Tabel 5.1 Data Video yang Digunakan	57
Tabel 5.2 Hasil pengujian PSNR pada video pesawat.avi.	59
Tabel 5.3 Hasil pengujian pada video car.avi dengan segmentasi ROI 86x56 piksel.....	61
Tabel 5.4 Waktu komputasi pada video pesawat.avi.....	62
Tabel 5.5 Hasil Pengujian waktu komputasi pada video car.avi	63

“Halaman ini sengaja dikosongkan”

DAFTAR GAMBAR

	Halaman
Gambar 2.1	Proses <i>Sampling</i> dan Kuantisasi 10
Gambar 2.2	Taksonomi dari metode tracking 13
Gambar 2.3	Grafik PBIM..... 18
Gambar 3.1	Diagram metodologi penelitian 23
Gambar 3.2	Diagram proses tahap pengujian Super resolusi 24
Gambar 4.1	<i>Use Case Diagram</i> perangkat lunak Super resolusi. 26
Gambar 4.2	<i>Swimlane Diagram</i> perangkat lunak Super resolusi 27
Gambar 4.3	ROI 28
Gambar 4.4	Hasil Tracking 29
Gambar 4.5	Diagram Alir Sistem..... 32
Gambar 4.6	Diagram Alir Proses <i>Tracking</i> dengan 2D-NCC 39
Gambar 4.7	Diagram Alir Proses Registrasi Citra Dengan PBIM 43
Gambar 4.8	Diagram Alir Proses Rekonstruksi Citra Dengan POCS 46
Gambar 4.9	Antar muka halaman awal 49
Gambar 4.10	Antar muka halaman proses <i>Tracking</i> dan Super resolusi..... 50
Gambar 4.11	Antar muka halaman proses <i>pengujian</i> Super resolusi 50
Gambar 4.12	Antar muka input video 51
Gambar 4.13	Antar muka pemilihan area ROI dengan Kotak warna biru pada <i>frame</i> tersebut adalah area ROI yang dipilih..... 52
Gambar 4.14	Tampilan untuk memasukkan data citra sampel 54
Gambar 4.15	delta_est merupakan nilai perpindahan (<i>displacement</i>) objek pada kordinat (x,y) untuk 20 citra sampel 55
Gambar 4.16	Citra Super Resolusi..... 56
Gambar 5.1	Citra objek hasil segmentasi dengan ROI 58
Gambar 5.2	Rangkaian Citra Hasil Tracking Resolusi Rendah pada ROI badan pesawat 58
Gambar 5.3	Super resolusi badan pesawat Garuda Indonesia 59

Gambar 5.4	Rangkaian citra hasil tracking resolusi rendah. car.avi	60
Gambar 5.5	Super resolusi plat nomor mobil	60
Gambar 5.6	Rangkaian Citra Hasil Tracking Resolusi Rendah pesawat	61
Gambar 5.7	Hasil Super resolusi objek pesawat.....	61
Gambar 5.8	Grafik perbandingan waktu komputasi pada pengujian video pesawat.avi.....	64
Gambar 5.9	Grafik perbandingan waktu komputasi pada pengujian video car.avi.....	64

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi telah semakin maju, baik dalam bidang *hardware* maupun *software*. Banyak perusahaan asing maupun dalam negeri yang memproduksi dan menjual produk elektronik berbasis multimedia seperti *smarthphone*, *notebook*, *tablet*, televisi, kamera, dan lain sebagainya. Perkembangan industri multimedia ini tidak terlepas dari citra sebagai salah satu parameter penting dalam kualitas teknologi multimedia. Selain itu, kebutuhan akan citra dimanfaatkan dalam bidang medis sebagai media pembantu diagnosa penyakit, peningkatan resolusi citra satelit, pengenalan pola, sistem keamanan berbasis citra, dan media pembelajaran visual yang efektif dalam bidang pendidikan. Dalam bidang-bidang tersebut citra dengan resolusi tinggi sangat dibutuhkan, sebab semakin tinggi resolusi citra maka informasi yang diperoleh dari citra tersebut lebih detail, sehingga analisis terhadap citra menjadi lebih akurat [2].

Super resolusi merupakan teknik untuk mendapatkan citra beresolusi tinggi dari citra yang beresolusi rendah[6]. Citra resolusi rendah yang digunakan dapat berupa citra tunggal maupun rangkaian citra yang diambil dari *scene* yang sama agar citra tersebut menyediakan informasi yang sama untuk proses rekontruksi citra resolusi tinggi. Pada dasarnya super resolusi memiliki dua tahapan, yaitu proses registrasi dan proses rekonstruksi.

Rangkaian citra biasanya diperoleh dari rangkaian *frame* tanpa adanya segmentasi menggunakan ROI (*Region Of Interest*). Super resolusi pada rangkaian *frame* ini umumnya dilakukan pada keseluruhan piksel *frame*, padahal kebutuhan informasi dari suatu citra hanya terdapat pada suatu bagian tertentu dari *frame* tersebut bukan pada keseluruhan piksel *frame*. Oleh karena itu, dibutuhkan cara untuk mendapatkan bagian *frame* tersebut agar proses super

resolusi dapat lebih efektif yaitu menggunakan ROI (*Region Of Interest*) dan melakukan proses *tracking* dalam rangkaian citra.

Dalam beberapa bidang video *tracking* dimanfaatkan sebagai media pengambilan informasi yang efektif. Misal pemanfaatan cctv untuk keamanan rumah, cctv pengawas lalu lintas dalam sistem *National Traffic Management Centre* (NTMC) sebagai pusat kendali informasi dan komunikasi yang mengatur lalu lintas Indonesia di Pulau Jawa, Sumatera, dan Bali pada titik rawan macet dan gangguan keamanan yang termonitor 24 jam[10], *Intelligent Transport System* (ITS) yang tujuan terpentingnya adalah mengawasi pelanggaran lalu lintas dan kejahatan di sarana transportasi umum berbasis video[8], dan lain sebagainya. Berdasarkan latar belakang diatas maka dalam penelitian tugas akhir ini proses *tracking* pada video akan dimanfaatkan untuk super resolusi sehingga resolusi citra pada bagian tertentu dari objek bergerak dapat ditingkatkan.

Beberapa penelitian sebelumnya yang terkait dengan penelitian tugas akhir ini adalah penelitian dengan metode POCS yang dilakukan oleh Sun Hao, Luo Lin, Zhou Weiping, dan Luo Limin pada tahun 2009 dengan judul “Location and Super-resolution Enhancement of License Plates Based on Video Sequences” dengan memanfaatkan metode rekonstruksi *Projection Onto Convex Sets* (POCS) dalam pengenalan plat nomor kendaraan bergerak[8]. Penelitian oleh Achmad Bryandi pada tahun 2010 dengan judul “Rekonstruksi Citra Pada Super Resolusi Menggunakan *Projection Onto Convex Sets*” yang menggunakan citra tunggal sebagai input[1]. Selain itu, penelitian super resolusi juga dilakukan oleh Budi Setiyono dkk. Pada penelitian ini metode PBIM dikolaborasikan dengan dua metode rekonstruksi yaitu POCS dan SANC[3]. Penelitian lain terkait objek *tracking* dilakukan oleh D. Hari Hara Santosh dkk yang berjudul *Tracking Multiple Moving Object Using Gaussian Mixture Model*. Dalam penelitian ini, metode GMM dan tracking analysis blob digunakan untuk mendeteksi bus yang lewat pada suatu jalan raya[13]. Berdasarkan beberapa penelitian tersebut,

PBIM merupakan metode registrasi yang baik dalam proses super resolusi dan POCS merupakan metode yang mampu merekonstruksi citra plat kendaraan pada objek kendaraan bergerak dengan baik.

Oleh karena itu pada tugas akhir ini akan diimplementasikan metode *Phase Based Image Matching* (PBIM) sebagai proses registrasi dan *Projection Onto Convex Sets Algorithm* (POCS) sebagai proses rekonstruksi. Citra objek dalam *frame* akan dipilih menggunakan ROI. Selanjutnya proses *tracking* pada tiap *frame* citra dilakukan untuk mendapatkan posisi objek pada serangkaian *frame* video. Hasil dari proses *tracking* berupa rangkain *frame* objek yang telah tersegmentasi digunakan sebagai input dalam proses registrasi menggunakan metode *Phase Based Image Matching* (PBIM). Hasil dari proses registrasi ini akan direkonstruksi menggunakan metode *Projection Onto Convex Sets Algorithm* (POCS) sehingga resolusi dari citra yang dipilih dalam ROI dapat meningkat serta menguji hasilnya.

1.2 Rumusan Masalah

Rumusan masalah dari Tugas Akhir ini adalah :

1. Bagaimana melakukan *tracking* pada bagian tertentu dari obyek yang bergerak
2. Bagaimana melakukan registrasi pada rangkaian citra hasil *tracking* menggunakan metode PBIM
3. Bagaimana merekonstruksi citra untuk menghasilkan citra resolusi tinggi menggunakan POCS serta menguji hasilnya.

1.3 Batasan Masalah

Batasan Masalah yang akan dibahas dalam penelitian tugas akhir ini adalah sebagai berikut :

1. Super resolusi akan dilakukan pada bagian citra yang telah ditentukan melalui ROI (Region of Interest) secara manual
2. Citra yang digunakan sebagai input dalam tugas akhir ini adalah video *tracking* obyek.
3. Super resolusi dilakukan pada *single* objek dan citra dalam *grayscale*

4. *Software* yang digunakan dalam implementasi penelitian ini adalah MATLAB

1.4 Tujuan

Tujuan dari tugas akhir ini adalah :

1. Melakukan *tracking* pada bagian tertentu dari obyek yang bergerak
2. Melakukan registrasi pada rangkaian citra hasil *tracking* menggunakan metode PBIM
3. Merekonstruksi citra untuk menghasilkan citra resolusi tinggi menggunakan POCS serta menguji hasilnya.

1.5 Manfaat

Manfaat dari tugas akhir ini adalah membantu pihak-pihak dalam menganalisis citra dengan resolusi tinggi. Misal pengenalan plat nomor kendaraan bergerak, sehingga plat tersebut dapat dikenali lebih jelas.

1.6 Sistematika Penulisan Tugas Akhir

Sistematika penulisan didalam Tugas Akhir ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang pembuatan tugas akhir, rumusan dan batasan permasalahan yang dihadapi dalam penelitian tugas akhir, tujuan dan manfaat pembuatan tugas akhir dan sistematika penulisan tugas akhir.

BAB II KAJIAN TEORI

Bab ini menjelaskan tentang penelitian sebelumnya yang mengkaji metode registrasi citra, rekonstruksi citra, serta metode *tracking* objek. Selain itu, pada bab ini akan dijelaskan kajian teori dari referensi penunjang serta penjelasan permasalahan yang dibahas dalam tugas akhir ini, meliputi Pengertian Citra Digital, *Obyek Tracking*, *Region Of Interest*, *Peak Signal to Noise*

Ratio untuk pengukuran kualitas citra rekonstruksi, Pengertian Super Resolusi, Metode Phased Based Image Matching (PBIM), Dan Metode Projection Onto Convex Sets Algorithm.

BAB III METODOLOGI PENELITIAN

Bab ini berisi metodologi atau urutan pengerjaan yang dilakukan dalam menyelesaikan tugas akhir, meliputi studi literatur, pengumpulan data, analisis dan desain sistem, pembuatan program, uji coba dan evaluasi, hingga penulisan tugas akhir.

BAB IV PERANCANGAN DAN IMPLEMENTASI

Bab ini menjelaskan analisis video, perancangan gambaran umum sistem, perancangan proses algoritma, perancangan data, dan perancangan antar muka sistem. Sistem ini memiliki inputan berupa segmentasi citra menggunakan ROI yang selanjutnya dilakukan proses *tracking* video untuk menentukan posisi objek dalam *frame-frame* citra. Hasil dari proses tracking ini berupa citra objek yang tersegmentasi dan digunakan sebagai citra observasi. Citra observasi ini akan diambil sebanyak n frame citra sebagai masukan proses registrasi menggunakan metode PBIM. Dalam proses registrasi ini akan diperoleh parameter translasi yang selanjutnya akan menjadi masukan dalam proses rekonstruksi menggunakan POCS untuk mendapatkan citra super resolusi.

BAB V PENGUJIAN DAN PEMBAHASAN HASIL

Bab ini akan menampilkan hasil uji coba, nilai *peak signal to noise ratio* (PSNR), dan hasil pengujian waktu komputasi pada proses super resolusi pada citra *tracking* dan pada citra keseluruhan piksel frame. Hasil

pengujian inilah yang akan digunakan dalam perumusan kesimpulan dan saran.

BAB VI PENUTUP

Bab ini merupakan penutup, berisi tentang kesimpulan yang dapat diambil berdasarkan data yang ada dan saran yang selayaknya dilakukan bila tugas akhir ini dilanjutkan.

BAB II

TINJAUAN PUSTAKA

Pada bab ini dijelaskan menjelaskan tentang kajian teori dari referensi penunjang serta penjelasan permasalahan yang dibahas dalam tugas akhir ini, meliputi Penelitian sebelumnya terkait Tuga Akhir ini, Pengertian Citra Digital, *Obyek Tracking*, *Region Of Interest*, *Peak Signal to Noise Ratio* untuk pengukuran kualitas citra rekonstruksi, Pengertian Super Resolusi, Metode Phased Based Image Matching (PBIM) untuk proses registrasi, Dan Metode Projection Onto Convex Sets Algorithm untuk proses rekonstruksi citra.

Penelitian Sebelumnya

Penelitian terkait tugas akhir ini tentang objek *tracking* dilakukan oleh D. Hari Hara Santosh dkk yang berjudul *Tracking Multiple Moving Object Using Gaussian Mixture Model*. Dalam penelitian ini, metode *Gaussian Mixture Models* dan *tracking blob analysis* digunakan untuk mendeteksi 56 bus dari 61 bus yang lewat. Berdasarkan hasil tersebut penelitian ini memiliki tingkat akurasi sistem sebesar 91.8 % [13].

Penelitian terkait metode registrasi citra pada tugas akhir ini dilakukan oleh Budi Setiyono dkk. Pada penelitian ini PBIM yang digunakan untuk memperkirakan translasi tahap registrasi dan dikolaborasikan dengan dua metode rekonstruksi, yaitu POCS dan *Structure - Adaptive Normalized convolution algorithm* (SANC). Hasil penelitian ini menunjukkan bahwa kolaborasi PBIM sebagai metode registrasi dengan POCS memiliki PSNR rata-rata 32,12205 sedangkan PBIM dan SANC memiliki PSNR rata-rata 32.07325 [3].

Selain itu, penelitian tentang metode super resolusi untuk proses rekonstruksi dengan metode POCS pertama kali dilakukan oleh Star dan Oskui [7]. Pada dasarnya POCS memiliki gagasan utama sebagai metode untuk mengestimasi citra super resolusi dalam gambaran persimpangan batas himpunan convex tertutup

dan memperoleh hasilnya dengan proses iterasi. Beberapa penelitian sebelumnya yang terkait dengan tugas akhir ini adalah penelitian dengan metode POCS yang dilakukan oleh Sun Hao, Luo Lin, Zhou Weiping, dan Luo Limin pada tahun 2009 dengan judul *Location and Super-resolution Enhancement of License Plates Based on Video Sequences* dengan memanfaatkan metode rekonstruksi *Projection Onto Convex Sets* (POCS) dalam pengenalan plat nomor kendaraan bergerak[8]. Penelitian oleh Achmad Bryandi pada tahun 2010 dengan judul *Rekonstruksi Citra Pada Super Resolusi Menggunakan Projection Onto Convex Sets* yang menggunakan citra tunggal sebagai input dan menghasilkan PSNR rata-rata sebesar 33.74837 dB[1].

Pada sub bab selanjutnya akan dijelaskan dasar-dasar teori super resolusi.

2.1 Pengertian Citra

Citra menurut Kamus Besar Bahasa Indonesia memiliki makna rupa, gambar, atau gambaran. Sedangkan menurut kamus Webster citra adalah suatu representasi, kemiripan, atau imitasi dari suatu objek atau benda. Citra terbagi menjadi dua yaitu citra diam dan citra bergerak. Citra diam adalah citra tunggal yang tidak bergerak. Sedangkan, citra bergerak adalah rangkaian citra diam yang ditampilkan secara beruntun sehingga memberi kesan pada mata kita sebagai gambar yang bergerak.

Dalam beberapa masa, citra yang dikenal manusia berbentuk citra kontinu. Suatu representasi objek yang dihasilkan dari sistem optik yang menerima sinyal analog dan dinyatakan dalam bidang dua dimensi. Nilai cahaya yang ditransmisikan pada citra kontinu memiliki rentang nilai yang tak terbatas. Contoh dari citra kontinu adalah mata manusia dan kamera analog.

Sebuah citra analog tidak dapat direpresentasikan secara langsung oleh komputer. Oleh sebab itu dilakukan sebuah proses untuk merubah nilai-nilai yang ada pada citra analog agar komputer dapat membaca dan menerjemahkan informasi yang

terdapat pada citra analog. Hasil dari pemrosesan tersebut dinamakan sebagai citra digital[9].

2.2 Citra Digital

Citra digital merupakan fungsi dua dimensi yang dapat dinyatakan dengan fungsi $f(x,y)$, dimana x dan y merupakan titik koordinat spasial. Dan amplitudo dari fungsi f pada sembarang koordinat (x,y) merupakan nilai intensitas cahaya, yang merupakan representasi dari warna cahaya yang ada pada citra analog. Citra digital adalah suatu citra dimana (x,y) dan nilai intensitas dari f terbatas (*discrete quantities*), dan telah dilakukan proses digitalisasi spasial dan digitalisasi kuantitas [4].

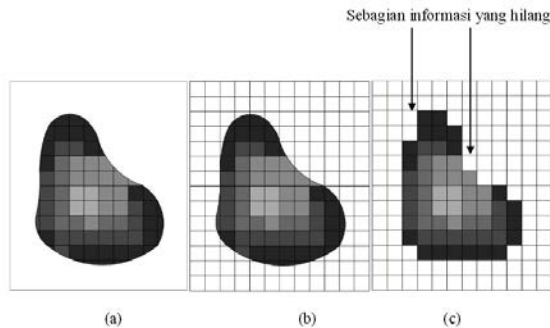
2.2.1 Digitalisasi Spasial (*Sampling*)

Sampling merupakan proses pengambilan informasi dari citra analog yang memiliki panjang dan lebar tertentu untuk membaginya ke beberapa blok kecil. Blok-blok tersebut disebut sebagai piksel. Sehingga citra digital yang lazim dinyatakan dalam bentuk matriks memiliki ukuran $M \times N$ dengan M sebagai baris dan N kolom. Bisa juga disebut sebagai citra digital yang memiliki $M \times N$ buah piksel. Notasi matriks citra digital dapat dinyatakan sebagai berikut:

$$f(x,y) = \begin{bmatrix} f(0,0) & \cdots & f(0, N-1) \\ \vdots & \ddots & \vdots \\ f(M-1,0) & \cdots & f(M-1, N-1) \end{bmatrix} \quad (2.1)$$

2.2.2 Digitalisasi Intensitas (Kuantisasi)

Kuantisasi adalah proses pemberian nilai derajat keabuan di setiap titik piksel yang merupakan representasi dari warna asli dari citra analog dengan Rentang nilai keabuan adalah 0 – 255.



Gambar 2.1. Proses *sampling* dan kuantisasi. a) Citra Digital, (b) Citra Digital Disampling Menjadi 14 Baris dan 12 Kolom, (c) Citra Digital Hasil Sampling Berukuran 14 x 12 Piksel[4].

2.3 Video Digital

Video adalah teknologi untuk menangkap, merekam, memproses, menyimpan, dan merekonstruksi suatu urutan dari beberapa citra. Alan C. Bovik dalam bukunya *Handbook of Image and Video Processing* menjelaskan bahwa video digital merupakan hasil *sampling* dan kuantisasi dari video analog[5]. Secara mendasar, tidak ada perbedaan proses *sampling* dan kuantisasi antara citra digital dan video digital.

Bagaimanapun juga, video analog yang kita lihat sehari-sehari seperti tampilan pada TV analog, sebenarnya bukan sesuatu yang benar-benar kontinu, melainkan terdiri dari beberapa *frame* yang ditampilkan dengan kecepatan tertentu. Setiap *frame* merupakan citra analog dan kecepatan untuk menampilkan citra-citra yang ada disebut sebagai *frame rate* dengan satuan *fps* (*frame per second*). Jika *frame rate* cukup tinggi, maka akan terlihat sebagai rangkaian yang kontinu sehingga tercipta ilusi gerak yang halus.

Video analog dapat dinyatakan dengan fungsi $I(x,y,t)$, dimana (x,y) adalah nilai kontinu dari fungsi I dan t menyatakan waktu. Sebenarnya tampilan video analog di TV maupun monitor merupakan representasi dari fungsi sinyal elektrik satu dimensi

$V(t)$. Dimana sinyal elektrik satu dimensi tersebut terdiri dari beberapa citra analog $I(x,y,t)$ dengan jumlah citra (x,y) tertentu dan waktu (t) tertentu. Proses pemisahan video ke beberapa unit *frame* citra disebut sebagai *scanning*.

2.4 Resolusi Citra

Resolusi citra adalah ukuran detail terkecil pada suatu citra. Dalam pengolahan citra digital ada dua klasifikasi paling umum yang sering digunakan dalam penelitian yakni resolusi spasial dan tingkat bit.

Resolusi spasial mengacu pada detail informasi yang dapat dilihat pada citra. Semakin tinggi resolusi sebuah citra akan semakin detail citra tersebut. Resolusi spasial pada dasarnya menggambarkan jumlah piksel dalam citra. Misal citra dengan ukuran 320 x 210 merepresentasikan citra dengan jumlah piksel sebanyak 320 pada kolomnya dan 210 pada barisnya[1].

Tingkat bit mengacu pada jumlah bit yang digunakan untuk menentukan warna pada setiap piksel suatu citra[1].

2.5 Citra Berwarna dan Citra Grayscale

Citra berwarna atau citra RGB (*Red- Green-Blue*) merupakan warna dasar yang dapat diterima oleh mata manusia. Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari ketiga warna dasar RGB. Setiap titik pada citra warna membutuhkan data sebesar 3 byte. Setiap warna dasar memiliki intensitas tersendiri dengan nilai minimum nol (0) dan nilai maksimum 255 (8 bit). RGB didasarkan pada teori bahwa mata manusia peka terhadap panjang gelombang 630nm (merah), 530 nm (hijau), dan 450 nm (biru).

Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, artinya nilai dari *Red = Green = Blue*. Nilai-nilai tersebut digunakan untuk menunjukkan intensitas warna. Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan warna putih pada

intensitas terkuat. Citra *grayscale* berbeda dengan citra "hitam-putih", dimana pada konteks komputer, citra hitam putih hanya terdiri atas 2 warna saja yaitu "hitam" dan "putih" saja. Pada citra grayscale warna bervariasi antara hitam dan putih, tetapi variasi warna diantaranya sangat banyak. Citra *grayscale* seringkali merupakan perhitungan dari intensitas cahaya pada setiap piksel pada spektrum elektromagnetik *single band*. Citra *grayscale* disimpan dalam format 8 bit untuk setiap sample piksel, yang memungkinkan sebanyak 256 intensitas.

Pada dasarnya resolusi citra secara matematis direpresentasikan dengan matriks dimana setiap nilai elemen matriksnya merupakan nilai dari intensitas warna baik itu dalam RGB atau dalam Grayscale.

2.6 Transformasi Fourier Diskrit Dua Dimensi

Secara umum, proses pengolahan citra digital dibagi dalam dua domain proses yaitu domain spasial dan domain frekuensi. Pada domain spasial proses pengolahan dilakukan dengan memanipulasi piksel citra secara langsung. Sedangkan pada domain frekuensi, proses yang dilakukan merupakan penyelesaian masalah tertentu yang sulit dilakukan pada domain spasial. Pengubahan somain spasial ke domain frekuensi dilakukan dengan cara transfor masi fourier.[1]

Dalam pengolahan citra, tranformasi fourier yang digunakan merupakan transformasi fourier diskrit. Sebab fungsi dari citra merupakan fungsi diskrit. Secara umum transformasi fourier diskrit dirumuskan dalam persamaan :

$$F(k_1, k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f(n_1, n_2) e^{-i 2\pi (\frac{k_1 n_1}{N_1} + \frac{k_2 n_2}{N_2})}$$

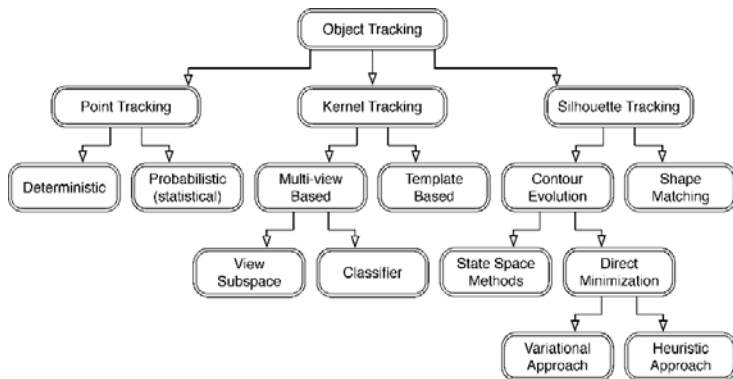
Setelah proses dalam domain frekuensi telah selesai dilakukan, maka citra dikembalikan ke domain spasial untuk mengetahui hasil berdasarkan nilai pikselnya menggunakan invers transformasi fourier.

$$f(n_1, n_2) = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} F(k_1, k_2) e^{i 2\pi \left(\frac{k_1 n_1}{N_1} + \frac{k_2 n_2}{N_2} \right)}$$

Dengan $f(n_1, n_2)$ adalah citra sebelum transformasi, $F(k_1, k_2)$ adalah citra hasil transformasi, N_1, N_2 ukuran citra dimana $\frac{1}{N_1 N_2}$ adalah nilai normalisasi citra pada invers transformasi fourier.

2.7 Obyek Tracking

Pelacakan objek merupakan hal penting dalam bidang *computer vision*. Proliferasi komputer bertenaga tinggi, ketersediaan kualitas tinggi, keterjangkauan harga kamera video, dan meningkatnya kebutuhan analisis video secara langsung telah menghasilkan banyak perkembangan dalam algoritma pelacakan objek. Ada tiga langkah utama dalam video Analisis: deteksi benda bergerak yang menarik, pelacakan benda tersebut dari bingkai frame, dan analisis objek trek untuk mengenali perilaku[11]. Representasi obyek, seleksi fitur *tracking*, deteksi obyek, *background subtraction*, segmentasi adalah hal-hal yang dilakukan dalam *tracking*. Akan tetapi, hal ini bergantung pada tingkat kebutuhan pengguna dalam mengambil informasi obyek pada citra *tracking*.



Gambar 2.2. Taksonomi dari metode *tracking*[11]

2.8 Template Matching

Template Matching merupakan salah satu metode dalam proses *tracking* yang digunakan untuk menemukan suatu bagian tertentu dari sebuah citra berdasarkan *template* yang telah ditentukan dan melakukan identifikasi sebagai bentuk pencocokan atau pengenalan citra objek pada *template* dengan citra lain yang serupa.

Pada penelitian tugas akhir ini metode *template matching* yang digunakan adalah metode *Normalized Cross Correlation*.

2.5.1 Metode Normalized Cross Correlation

Permasalahan utama pada *template matching* adalah untuk menentukan posisi dari bagian citra yang telah ditentukan dalam citra dua dimensi. Jika $f(x, y)$ menotasikan nilai intensitas dari citra f yang berukuran $M_x \times M_y$ pada titik (x, y) , $x \in \{0, \dots, M_x - 1\}$ dan $y \in \{0, \dots, M_y - 1\}$. Sedangkan *template* t yang diberikan berukuran $N_x \times N_y$. Selanjutnya untuk menentukan posisi hasil pencarian adalah dengan melakukan penghitungan nilai *normalized cross correlation*, γ , pada tiap titik (u, v) untuk citra f dan *template* t menggunakan persamaan :

$$\gamma = \frac{\sum_{x,y} (f(x, y) - \bar{f}_{u,v})(t(x - u, y - v) - \bar{t})}{\sqrt{\sum_{x,y} (f(x, y) - \bar{f}_{u,v})^2 \sum_{x,y} (t(x - u, y - v) - \bar{t})^2}} \quad (2.2)$$

Pada persamaan (2) $\bar{f}_{u,v}$ dinotasikan sebagai nilai mean dari $f(x, y)$ dalam area dari *template* t yang bergeser ke (u, v) , dihitung dengan persamaan :

$$\bar{f}_{u,v} = \frac{1}{N_u N_y} \sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} f(x, y) \quad (2.3)$$

2.9 Region Of Interest (ROI)

Region Of Interest (ROI) adalah bagian dari suatu gambar yang ingin Anda menyaring atau melakukan beberapa operasi

lain. Mendefinisikan ROI dengan menciptakan tanda biner, yang merupakan citra biner yang merupakan ukuran yang sama dengan gambar yang dikelola dengan piksel yang mendefinisikan set ROI 1 dan semua piksel lainnya diatur ke 0. ROI dapat dipilih lebih dari satu dalam satu frame. Misalnya pemilihan daerah geografis, pengambilan poligon yang mencakup piksel berdekatan, atau didefinisikan oleh berbagai intensitas[12].

2.10 Peak Signal to Noise Ratio (PSNR)

PSNR merupakan metode pengukuran yang banyak digunakan untuk sistem kompresi dan rekonstruksi gambar. PSNR didefinisikan sebagai berikut :

$$PSNR = 20 \log_{10} \frac{255}{\sqrt{MSE}} \quad (2.4)$$

MSE adalah metode pengukuran kontrol dan kualitas yang diperoleh dari perhitungan perbandingan contoh obyek dengan obyek asli sehingga diketahui tingkat ketidaksesuaiannya. Persamaan MSE terhadap deviasi target dirumuskan sebagai berikut :

$$MSE = \frac{1}{MN} \sum_{y=1}^M \sum_{x=1}^N [I(x,y) - I'(x,y)]^2 \quad (2.5)$$

Dengan $I(x,y)$ adalah nilai piksel asli, sedangkan $I'(x,y)$ adalah nilai piksel citra hasil rekonstruksi, dan M, N adalah dimensi citra[3].

2.11 Super Resolusi

Super Resolusi merupakan teknik untuk mendapatkan citra yang beresolusi tinggi dari citra yang beresolusi rendah[6]. Resolusi yang dimaksud pada teknik Super Resolusi ini adalah resolusi spasial. Citra resolusi rendah yang digunakan dapat berupa citra tunggal atau rangkaian citra yang diambil dari *scene* yang sama. Karena dari *scene* yang sama itu menyediakan informasi yang dapat digunakan untuk merekonstruksi citra resolusi tinggi.

Pada umumnya, Super Resolusi terdiri dari dua tahap, yaitu registrasi citra dan rekonstruksi citra. Rangkaian citra resolusi rendah yang digunakan pada Super Resolusi ini harus memiliki nilai pergeseran sampai tingkat sub-piksel, karena dari pergeseran ini maka setiap citra memiliki informasi yang berbeda-beda sehingga citra resolusi tinggi dapat dibentuk.

2.12 Metode *Phased Based Image Matching* (PBIM)

Berikut ini akan dijelaskan tentang metode registrasi menggunakan *Phased Based Image Matching* yang akan digunakan dalam sistem super resolusi. Untuk mempermudah pemahaman maka misalkan diberikan dua citra dengan $f(n_1, n_2)$ dan $g(n_1, n_2)$ dengan dimensi $N_1 \times N_2$. Asumsikan indeks n_1 berkisar dari $-M_1, \dots, M_1$ dan indeks n_2 berkisar dari $-M_2, \dots, M_2$. Untuk lebih sederhananya digunakan persamaan indeks n_1 berkisar dari $N_1 = 2M_1 + 1$ dan $N_2 = 2M_2 + 1$ [3].

Citra digital adalah suatu citra dimana (x, y) dan nilai intensitas dari f terbatas (*discrete quantities*), dan telah dilakukan proses digitalisasi spasial dan digitalisasi kuantitas [4]. Oleh karena itu fungsi dari citra merupakan fungsi diskrit sehingga digunakan Transformasi Fourier Diskrit. Transformasi ini dibuat untuk fungsi diskrit $f(x, y)$ yang tidak bernilai nol pada daerah $0 \leq x \leq M - 1$ dan $0 \leq y \leq N - 1$ [1]. Persamaan Transformasi Fourier Diskrit tersebut diberikan sebagai berikut :

$$\begin{aligned} F(k_1, k_2) &= \sum_{n_1 n_2} f(n_1, n_2) W_{N_1}^{k_1 n_1} W_{N_2}^{k_2 n_2} \\ &= A_F(k_1, k_2) e^{j\theta_F(k_1, k_2)} \end{aligned} \quad (2.6)$$

$$\begin{aligned} G(k_1, k_2) &= \sum_{n_1 n_2} g(n_1, n_2) W_{N_1}^{k_1 n_1} W_{N_2}^{k_2 n_2} \\ &= A_G(k_1, k_2) e^{j\theta_G(k_1, k_2)} \end{aligned} \quad (2.7)$$

dengan :

1. $F(k_1, k_2)$ dan $G(k_1, k_2)$ menunjukkan Transformasi Fourier Diskrit dari domain spasial citra $f(n_1, n_2)$ dan $g(n_1, n_2)$
2. n_1, n_2 adalah elemen index domain spasial pada $f(n_1, n_2)$ sedangkan k_1, k_2 adalah elemen index domain frekuensi pada $F(k_1, k_2)$ dimana $k_1 = -M_1, \dots, M_1$ dan $k_2 = -M_2, \dots, M_2$
3. $A_F(k_1, k_2)$ dan $A_G(k_1, k_2)$ adalah komponen amplitudo
4. $e^{j\theta_F(k_1, k_2)}$ dan $e^{j\theta_G(k_1, k_2)}$ adalah komponen fase
5. Operator \sum_{n_1, n_2} menunjukkan $\sum_{n_1=-M_1}^{M_1} \sum_{n_2=-M_2}^{M_2}$
6. $W_{N_1} = e^{\frac{j2\pi}{N_1}}$, $W_{N_2} = e^{\frac{j2\pi}{N_2}}$ dimana j adalah komponen imajiner

Fase spektrum silang (Cross Spectrum Phase)

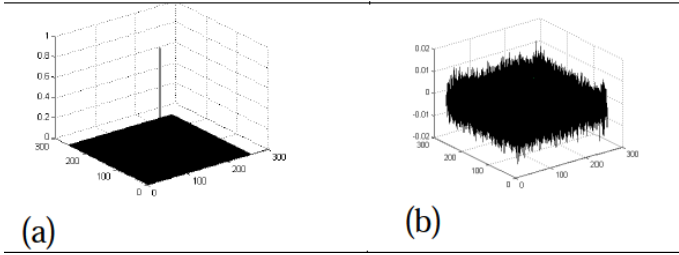
$\hat{R}(k_1, k_2)$ didefinisikan dalam persamaan sebagai berikut :

$$\hat{R}(k_1, k_2) = \frac{F(k_1, k_2)\overline{G(k_1, k_2)}}{|F(k_1, k_2)\overline{G(k_1, k_2)}|} = e^{j\theta(k_1, k_2)} \quad (2.8)$$

Invers Phase-Based 2D dari fungsi $\hat{R}(k_1, k_2)$ ditunjukkan dalam persamaan :

$$\hat{r}(n_1, n_2) = \frac{1}{N_1 N_2} \sum_{k_1, k_2} \hat{R}(k_1, k_2) W_{N_1}^{k_1 n_1} W_{N_2}^{k_2 n_2} \quad (2.9)$$

Hasil yang diperoleh dari proses registrasi menggunakan PBIM ini adalah skala translasi piksel yang selanjutnya akan digunakan dalam proses rekonstruksi menggunakan POCS.



Gambar 2.3. Grafik PBIM (a) Citra identik
(b) Citra non identik[3].

2.13 Metode *Projection Onto Convex sets Algorithm* (POCS)

Setelah mendapatkan hasil skala dan parameter translasi maka diimplementasikan dalam metode Projection onto convex sets (POCS) untuk merekonstruksi dan meningkatkan resolusi citra. Konsep utama sebagai metode untuk mengestimasi citra super resolusi dalam rangkaian citra yang dibatasi dalam himpunan convex tertutup dan memperoleh hasilnya melalui proses iterasi[8].

Hal ini dapat diterapkan untuk menemukan vektor yang termasuk dalam persimpangan dengan formulasi rekursi :

$$x^{(n+1)} = P_m P_{m-1} \dots P_2 P_1 x^{(n)}$$

Dengan $x^{(n)}$ adalah estimasi super resolusi setelah iterasi ke-n, $x^{(0)}$ adalah iterasi pertama dari citra HR yang merupakan perbesaran citra dengan algoritma interpolasi. P_m adalah operator proyeksi yang memproyeksikan citra ke himpunan convex tertutup.

Constraint set (himpunan kendala) untuk setiap piksel didalam rangkaian citra *Low Resolution* (*LR*) $g(m_1, m_2, l)$ didefinisikan sebagai :

$$C_{m_1, m_2, l} = \{ f(n_1, n_2, k): |r^{(x)}(m_1, m_2, l)| \leq \delta_0(m_1, m_2, l) \} \quad (2.10)$$

Dimana

$$r^{(x)}(m_1, m_2, l) = g(m_1, m_2, l) - \sum_{n_1, n_2} f(n_1, n_2, k) \cdot h(m_1, m_2; n_1, n_2, l) \quad (2.11)$$

$\delta_0(m_1, m_2, l)$ adalah *noise*. $h(m_1, m_2; n_1, n_2, l)$ adalah pergeseran linear dari fungsi persebaran titik (*point spread function*).

Operator Proyeksi dari tiap piksel $x(n_1, n_2, k)$ terhadap $C_{m_1, m_2, l}$ didefinisikan sebagai :

$$P_{m_1, m_2, l}[x(n_1, n_2, k)] = \begin{cases} x + \frac{r^{(x)} - \delta_0}{H} h, & r^{(x)} > \delta_0 \\ x, & -\delta_0 \leq r^{(x)} \leq \delta_0 \\ x + \frac{r^{(x)} + \delta_0}{H} h, & r^{(x)} < -\delta_0 \end{cases} \quad (2.12)$$

Dimana : $H = \sum_{n_1} \sum_{n_2} h^2(m_1, m_2; n_1, n_2, l)$

$h = h(m_1, m_2; n_1, n_2, l)$

$x = x(n_1, n_2, k)$

$r = r^{(x)}(m_1, m_2, l)$

$\delta_0 = \delta_0(m_1, m_2, l)$

Setelah mendapatkan nilai awal interpolasi, $r^{(x)}(m_1, m_2, l)$ akan dihitung berdasarkan persamaan (9) untuk setiap piksel $x(n_1, n_2, k)$. Lalu akan diproyeksikan balik dengan $P_{m_1, m_2, l}$ pada persamaan (10). Jika kriteria pemberhentian diperoleh, iterasi akan berhenti, sebaliknya maka iterasi akan terus berlanjut[8].

“Halaman ini sengaja dikosongkan”

BAB III METODOLOGI

Bab ini membahas mengenai metodologi sistem yang digunakan untuk menyelesaikan tugas akhir. Pembahasan metodologi sistem diawali dengan penjelasan tentang objek penelitian, peralatan yang digunakan, dan tahap penelitian.

3.1 Objek Penelitian

Objek penelitian yang akan digunakan pada tugas akhir adalah video rekaman obyek bergerak yang dikonversi ke dalam citra *grayscale*.

3.2 Peralatan

Peralatan penelitian yang digunakan untuk menyelesaikan tugas akhir ini adalah MATLAB sebagai perangkat lunak utama untuk membuat antarmuka dan membangun sistem super resolusi.

3.3 Tahap Penelitian

Adapun tahap-tahap yang dilakukan dalam penyusunan Tugas Akhir ini adalah sebagai berikut :

1. Studi Literatur

Pada tahap ini akan dikaji tentang pengolahan video *tracking* dan konsep super resolusi yang diimplementasikan pada video *tracking* tersebut. Studi ini dilakukan dengan membaca jurnal video *tracking*, jurnal super resolusi yang berkaitan dengan penggunaan metode *Phased Based Image Matching* dan *Project Onto Convex Sets Algorithm*.

2. Analisis dan Perancangan Sistem

Pada tahapan ini akan dilakukan analisis video, perancangan gambaran umum sistem, perancangan proses algoritma, perancangan data, dan perancangan antar muka sistem. Sistem ini memiliki inputan berupa segmentasi citra menggunakan

ROI yang selanjutnya dilakukan proses *tracking* video untuk menentukan posisi objek dalam *frame-frame* citra. Hasil dari proses tracking ini berupa citra objek yang tersegmentasi dan digunakan sebagai citra observasi. Citra observasi ini akan diambil sebanyak n frame citra sebagai masukan proses registrasi menggunakan metode PBIM. Dalam proses registrasi ini akan diperoleh parameter translasi yang selanjutnya akan menjadi masukan dalam proses rekonstruksi menggunakan POCS untuk mendapatkan citra super resolusi. Diagram proses Super resolusi dapat dilihat pada gambar 3.1.

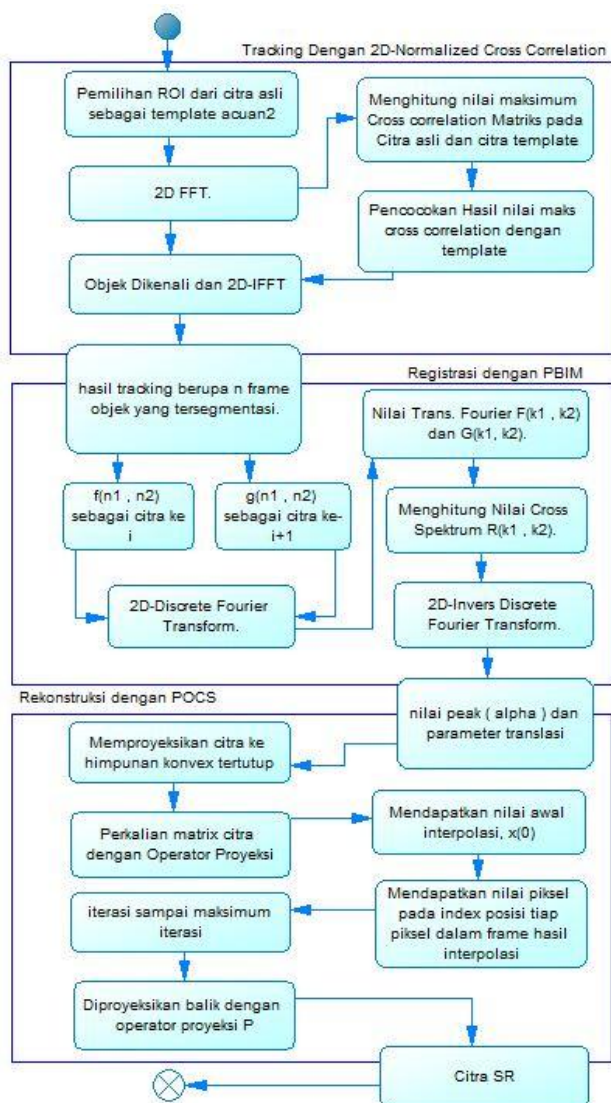
3. Pengujian dan Evaluasi Sistem

Pada tahap ini akan dilakukan pengujian terhadap program yang telah dibangun. Pengujian ini juga akan dihitung nilai PSNR agar diperoleh nilai representasi tingkat keberhasilan program. Diagram uji dapat dilihat pada gambar 3.2.

4. Penarikan Kesimpulan

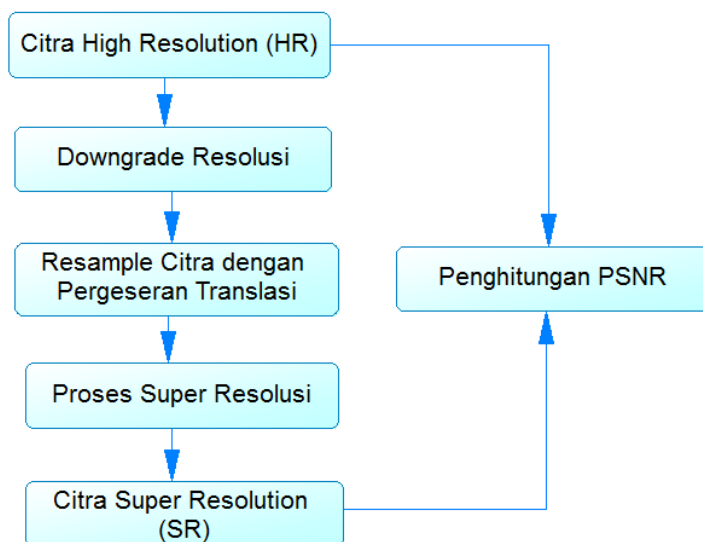
Tahap penarikan kesimpulan merupakan tahap akhir dalam proses penelitian Tugas Akhir ini, dimana pada tahap ini dilakukan penarikan kesimpulan terhadap hasil yang telah dicapai.

Tahap-tahap pengerjaan Tugas Akhir yang telah dijelaskan di atas digambarkan dalam diagram alir pada Gambar 3.1.



Gambar 3.1. Diagram Metodologi penelitian

Berikut merupakan Diagram tahap Pengujian Super Resolusi :



Gambar 3.2. Diagram Proses Tahap Pengujian Super Resolusi

BAB IV

PERANCANGAN DAN IMPLEMENTASI SISTEM

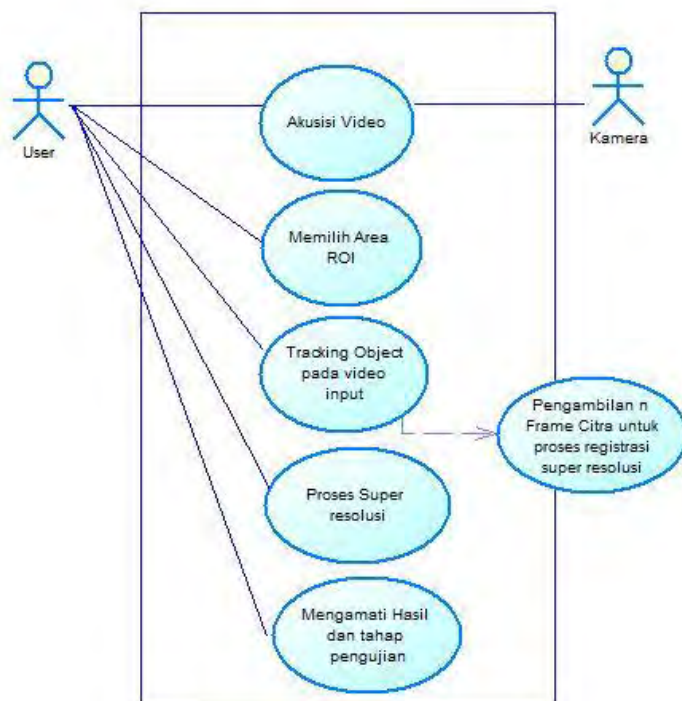
Pada bab ini akan dibahas mengenai perancangan dan implementasi sistem dimulai dari pembahasan proses akuisisi citra, *tracking* citra menggunakan metode *2D Normalized Cross Correlation*, proses super resolusi yang meliputi proses registrasi menggunakan metode *phased based image matching* dan proses rekonstruksi menggunakan metode *projection onto convex sets algorithm*, serta penjelasan mengenai cara untuk mendapatkan data keluaran yang sesuai dengan tujuan dari penelitian Tugas Akhir ini.

4.1 Analisis Sistem

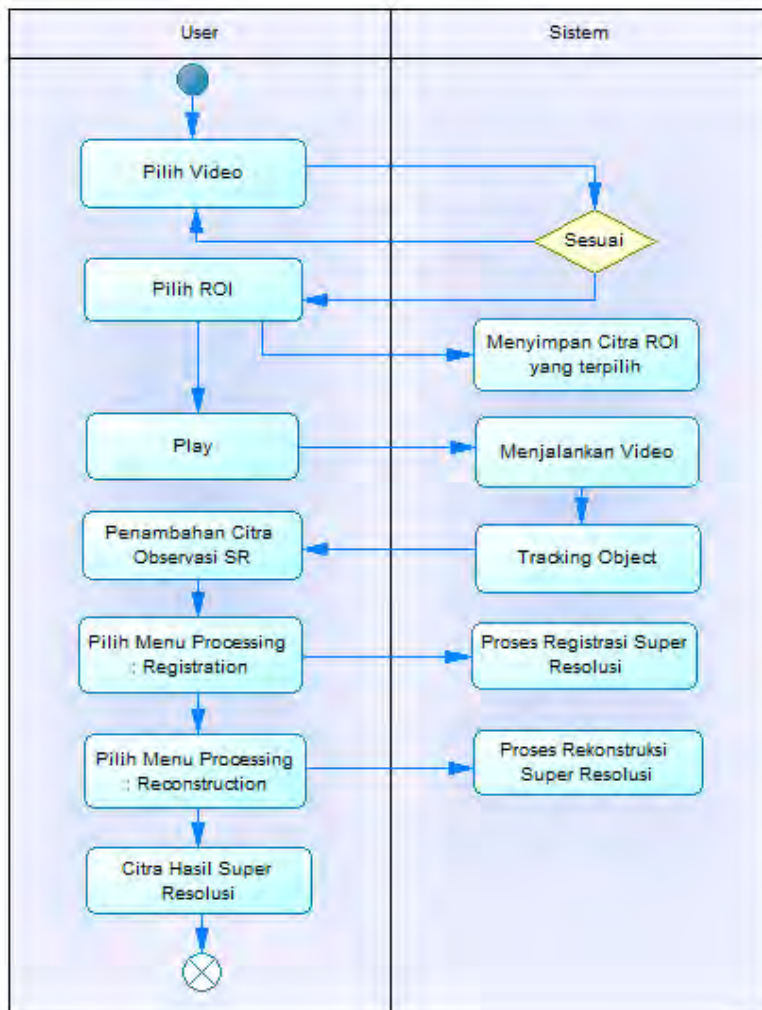
Dalam sub bab ini akan dijelaskan proses analisis dalam membangun perangkat lunak super resolusi, mulai dari *Use Case Diagram*, *Activity Diagram*, proses akuisisi citra, segmentasi citra, *tracking object* pada video input, proses super resolusi yang meliputi proses registrasi menggunakan metode *phased based image matching (PBIM)* dan proses rekonstruksi menggunakan metode *projection onto convex sets algorithm (POCS)*, serta pengujian hasil dari sistem.

4.1.1 Analisis Sistem Perangkat Lunak

Perangkat lunak yang akan dibangun dapat membantu pihak-pihak dalam menganalisis citra dengan resolusi tinggi pada bagian tertentu dari rangkaian frame yang dipilih dengan ROI. Misal analisis dan pengenalan plat nomor kendaraan bergerak, sehingga plat tersebut dapat dikenali lebih jelas, analisis citra satelit, diagnosa penyakit, sistem keamanan berbasis citra digital, dan sebagainya. Secara umum *Use Case Diagram* dari sistem ini disajikan pada Gambar 4.1, *Swimlane Diagram* pada Gambar 4.2 menunjukkan jalannya sistem perangkat lunak.



Gambar 4.1. *Use Case Diagram* Perangkat Lunak Super Resolusi



Gambar 4.2. *Swimlane Diagram* Perangkat Lunak Super Resolusi.

Sistem perangkat lunak yang dibangun ini memiliki beberapa tahapan sebagai berikut :

a. Akuisisi video

Data masukan berupa rekaman video digital *offline* yang kemudian di-*scanning*. *Scanning* adalah proses pemecahan video menjadi beberapa rangkaian citra yang sering disebut dengan *frame*.

Data video *input* tersebut harus memenuhi kriteria :

- Video input hanya memuat *single* objek dan dikonversi kedalam citra *grayscale*.
- Video yang digunakan memiliki *framerate* minimal 25 *frame per second* (fps).

b. Penentuan ROI (*Region of Interest*)

ROI (*Region of Interest*) akan digunakan sebagai area segmentasi atau pemilihan objek yang akan tersimpan sebagai template acuan *tracking object*. Berdasarkan ROI ini user dapat memilih objek sesuai kebutuhan, dengan kata lain piksel-piksel selain dalam area ROI tidak akan terpilih.



Gambar 4.3. ROI (a) Citra dengan ROI (daerah didalam kotak biru) (b) Penentuan Posisi objek pada *frame* pertama

c. *Tracking Object*

Tracking object ini dilakukan berdasarkan objek yang terpilih dalam ROI sebagai *template* dasar. Selanjutnya metode 2D *Normalized Cross Correlation* diimplementasikan untuk mendeteksi perpindahan objek[14][15]. *Tracking* dengan metode 2D-NCC ini bertujuan bahwa objek yang terdeteksi pada *frame* ke-*n* dengan *frame* ke-*n*+1 merupakan objek yang sama. Lokasi dari objek ditentukan dengan mencari nilai maksimum

Normalized Cross Correlation. Untuk proses lebih detail akan dibahas pada sub bab 4.2.3.1

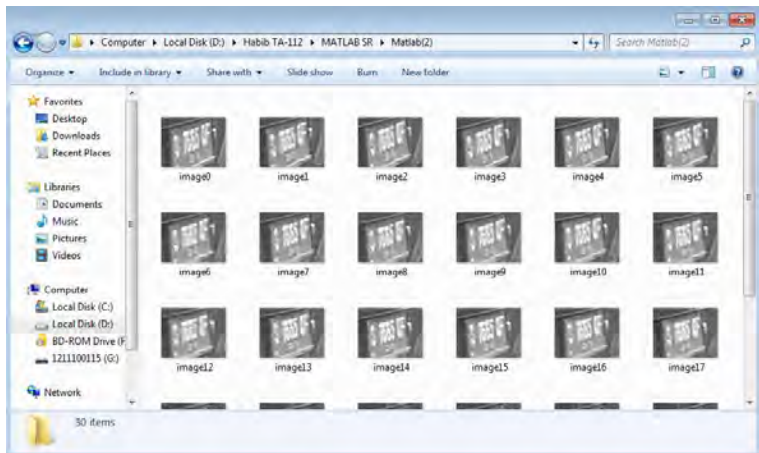
Target yang telah teridentifikasi akan ditandai dengan *bounding box*. Setelah itu, citra didalam *bounding box* akan disegmentasi dan disimpan sebagai bahan masukan proses registrasi super resolusi

d. Penyimpanan Citra Objek Hasil Tracking

Penyimpanan citra ini merupakan proses untuk menyimpan *frame* citra yang berhasil menentukan posisi objek. Selanjutnya hasil penyimpanan ini digunakan sebagai sampel dalam proses registrasi citra menggunakan PBIM.



(a)



(b)

Gambar 4.4. Hasil Tracking (a) Objek terdeteksi dalam proses *tracking* (b) merupakan data objek yang terdeteksi dan disimpan dalam direktori.

e. Proses registrasi super resolusi

Objek hasil *tracking* yang telah disimpan tersebut akan diambil n citra sebagai masukan proses registrasi super resolusi menggunakan metode *phase based image matching (PBIM)*. Metode ini memungkinkan untuk mengestimasi nilai perpindahan posisi (*displacement*) objek pada citra dengan akurasi sub piksel dari posisi puncak korelasi. Selain itu, sudut rotasi citra dan faktor skala dari pengurangan atau perbesaran citra dapat diketahui dengan mengkonversikannya menjadi nilai translasi citra. Oleh karena itu estimasi translasi perpindahan citra dengan akurasi yang tinggi sangat dibutuhkan [16]. Untuk proses lebih detail akan dibahas pada sub bab 4.2.3.2

f. Proses rekonstruksi super resolusi

Nilai parameter translasi dari proses registrasi diimplementasikan dalam metode *Projection Onto Convex Sets (POCS)* untuk merekonstruksi dan meningkatkan resolusi citra. Metode ini digunakan untuk mengestimasi citra super resolusi dari rangkaian citra yang dibatasi dalam himpunan convex tertutup dan memperoleh hasilnya melalui proses iterasi[8].

Vektor citra diperoleh dari formula rekursi :

$$x^{(n+1)} = P_m P_{m-1} \dots P_2 P_1 x^{(n)}$$

Dengan $x^{(n)}$ adalah estimasi super resolusi setelah iterasi ke-n, $x^{(0)}$ adalah iterasi pertama dari citra HR yang merupakan perbesaran citra dengan algoritma interpolasi. P_m adalah operator proyeksi. Untuk proses lebih detail akan dibahas pada sub bab 4.2.3.3

4.1.2 Analisis Kebutuhan Sistem

Perangkat lunak ini dibangun menggunakan software MATLAB R2013a baik dari desain interface, sistem *toolbox computer vision* dan pengolahan video digital.

Tabel 4.1. Tabel kebutuhan sistem

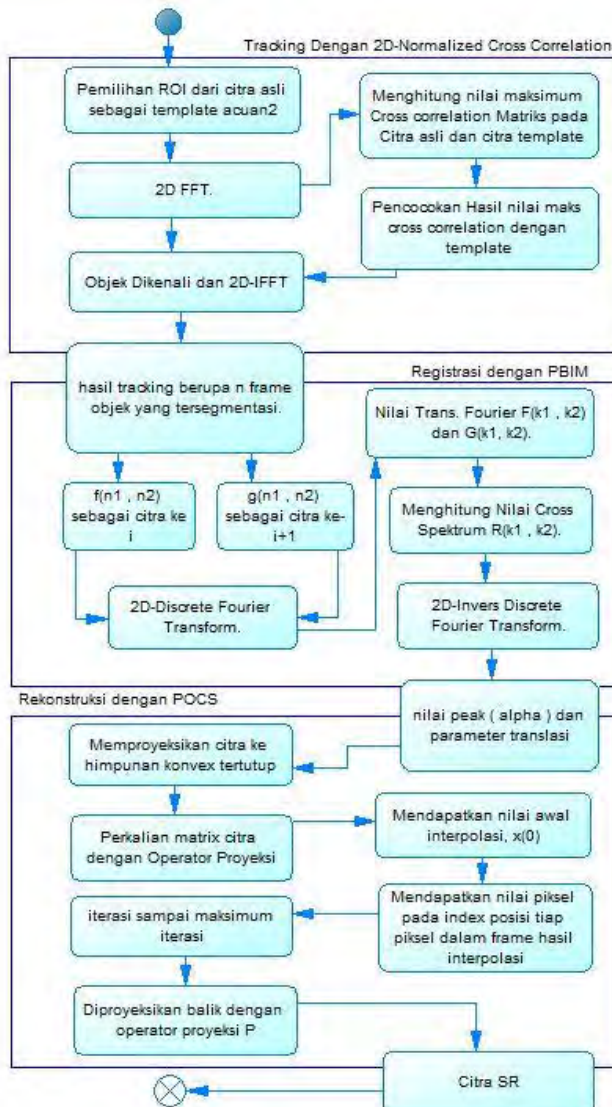
Perangkat Keras	Prosesor : Pentium(R) Dual-Core CPU T4300@2.10Ghz(2 CPUs),~2.1Ghz
	RAM Memory : 1,24 GB
Perangkat Lunak	Sistem Operasi : Windows 7 Ultimate 32-bit
	Tools : MATLAB R2013a

4.2 Perancangan Sistem

Setelah analisis sistem, kemudian dilanjutkan dengan perancangan sistem. Perancangan sistem tersebut meliputi perancangan data sistem, perancangan *class* sistem, perancangan proses algoritma sistem, dan perancangan antarmuka sistem.

4.2.1 Gambaran Sistem

Pada subbab ini akan disajikan diagram alir sistem secara umum berupa proses berjalannya sistem secara umum yang terbagi menjadi 3 tahapan yaitu tracking object dengan metode *2D-Normalized Cross Correlation*, registrasi citra dengan metode *Phased Based Image Matching*, dan rekonstruksi citra dengan metode *Projection Onto Convex Sets Algorithm*. Berikut diagram alir sistem yang disajikan pada gambar 4.5



Gambar 4.5. Diagram Alir Sistem

4.2.2 Penjelasan Umum Sistem

Proses algoritma ini dimulai dengan pemilihan ROI (*Region of Interest*) objek dalam frame video yang digunakan sebagai *template tracking*. Selanjutnya metode *2D-Normalized Cross Correlation* akan menentukan posisi-posisi objek pada tiap *frame* video yang ditandai dengan *bounding box*. Objek dalam *bounding box* akan disimpan menjadi citra gambar dengan format *jpg* pada *directory* yang sudah ditentukan. Citra-citra inilah yang akan menjadi input proses registrasi citra menggunakan metode *phase based image matching*. Proses registrasi ini akan menghasilkan nilai parameter translasi objek tiap *frame*. Selanjutnya, nilai parameter translasi tersebut digunakan dalam proses rekonstruksi menggunakan metode *projection onto convex sets algorithm* untuk mendapatkan hasil berupa citra super resolusi.

Pada sub bab selanjutnya akan dijelaskan algoritma tiap proses dalam metode yang digunakan.

4.2.3 Perancangan Proses Algoritma

Perancangan proses algoritma ini secara garis besar di bagi dalam tiga tahapan, yaitu proses *tracking*, registrasi citra, dan rekonstruksi citra. Pada awalnya citra dari *frame* video diambil area ROI secara manual. Selanjutnya dari citra ROI tersebut akan disimpan sebagai *template* acuan untuk proses *tracking*.

4.2.3.1 Perancangan Proses *Tracking*

Pada proses ini citra yang terpilih dalam ROI dan citra asli akan dikenai fast fourier transform (FFT). Selanjutnya persamaan :

$$\gamma = \frac{\sum_{x,y} (f(x,y) - \bar{f}_{u,v})(t(x-u, y-v) - \bar{t})}{\sqrt{\sum_{x,y} (f(x,y) - \bar{f}_{u,v})^2 \sum_{x,y} (t(x-u, y-v) - \bar{t})^2}} \quad (4.1)$$

digunakan untuk mengkonversi matriks citra menjadi matrik Cross correlation, $\gamma(u, v)$. Pada konteks ini fungsi matriks dari *template* yang terpilih juga dikonversi kedalam matriks *cross correlation*. Representasi dari citra dalam matriks cross correlation ini akan digunakan untuk pencarian dari posisi *template* pada tiap *frame* yang dilakukan dengan penghitungan nilai maksimum dari *cross correlation*. Pencocokan dilakukan dan posisi *template* diperoleh.

Implementasi dari algoritma *invers fast fourier transform* dilakukan pada *template* untuk mengembalikan matriks citra *template*. Hasil dari *tracking* yang tersegmentasi disimpan sebagai serangkaian *frame* citra objek. Jika $\sum f^2(x, y)$ berubah-ubah, maka penghitungan jarak kesamaan antara citra asli dan *template* akan gagal didapatkan. Selain itu, ukuran dari tingkat kesamaan citra dan *template* invarian terhadap perubahan amplitudo citra, seperti adanya perubahan pencahayaan selama rangkaian citra.

Contoh dari perhitungan transformasi fourier:

Misalkan diberikan

$$f(n_1, n_2) = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad g(n_1, n_2) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Transformasi fourier

$$F(k_1, k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f(n_1, n_2) e^{-2\pi i (\frac{k_1 n_1}{N_1} + \frac{k_2 n_2}{N_2})}$$

Maka, untuk $f(n_1, n_2)$ akan dihitung $F(k_1, k_2)$

$$F(k_1, k_2) = \sum_{n_1=0}^2 \sum_{n_2=0}^2 f(n_1, n_2) e^{-2\pi i (\frac{k_1 n_1}{3} + \frac{k_2 n_2}{3})}$$

$$F(0,0) = \sum_{n_1=0}^2 \sum_{n_2=0}^2 f(n_1, n_2) e^{-2\pi i (\frac{0n_1}{3} + \frac{0n_2}{3})}$$

$$\begin{aligned}
F(0,0) &= \sum_{n_1=0}^2 \sum_{n_2=0}^2 f(n_1, n_2) \\
&= f(0,0) + f(0,1) + f(0,2) + f(1,0) + f(1,1) \\
&\quad + f(1,2) + f(2,0) + f(2,1) + f(2,2) \\
&= 1 + 2 + 0 + 2 + 1 + 0 + 0 + 1 + 1 \\
&= 8 \\
F(0,1) &= \sum_{n_1=0}^2 \sum_{n_2=0}^2 f(n_1, n_2) e^{-2\pi i(\frac{n_2}{3})} \\
&= f(0,0)e^{-2\pi i 0} + f(0,1)e^{-2\pi i \frac{1}{3}} + f(0,2)e^{-2\pi i \frac{2}{3}} + \\
&\quad f(1,0)e^{-2\pi i 0} + f(1,1)e^{-2\pi i \frac{1}{3}} + f(1,2)e^{-2\pi i \frac{2}{3}} + \\
&\quad f(2,0)e^{-2\pi i 0} + f(2,1)e^{-2\pi i \frac{1}{3}} + f(2,2)e^{-2\pi i \frac{2}{3}} \\
&= 1 + 2(-0,5 - 0,86i) + 0 + 2 + 1(-0,5 - 0,086i) + \\
&\quad 1(-0,5 + 0,86i) + 0 + 0 + 1(-0,5 + 0,86i) \\
&= 0,5 - 0,86i
\end{aligned}$$

Dengan cara yang sama, maka didapatkan transformasi fourier diskrit dari matriks $f(n_1, n_2) = F(k_1, k_2)$.

$$F = \begin{bmatrix} 8 & 0,5 - 0,86i & 0,5 - 0,86i \\ 0,5 - 2,59i & -1 + 3,46i & 0,5 + 0,86i \\ 0,5 + 2,59i & 0,5 - 0,86i & -1 + 3,46i \end{bmatrix}$$

Dengan cara yang sama, maka didapatkan transformasi fourier diskrit dari matriks $g(n_1, n_2) = G(k_1, k_2)$.

$$G = \begin{bmatrix} 8 & -2,5 - 0,86i & -2,5 - 0,86i \\ 0,5 - 2,59i & -2,5 + 0,86i & 0,5 - 0,86i \\ 0,5 + 2,59i & 0,5 + 0,86i & -2,5 + 0,86i \end{bmatrix}$$

$$\hat{R}(k_1, k_2) = \frac{F(k_1, k_2) \cdot \overline{G(k_1, k_2)}}{|F(k_1, k_2) \cdot \overline{G(k_1, k_2)}|}$$

$\hat{R}(k_1, k_2)$ menyatakan masing-masing elemen dari matriks *cross spectrum*. Setelah selesai menghitung nilai dari setiap elemen akan didapatkan:

$$\hat{R} = \begin{bmatrix} 1 & 0,51 + 2,58i & -0,51 - 2,58i \\ 1 - 5,18i & -0,47 - 9,51i & 0,98 \\ 1 + 5,18i & 0,98 & 5,47 - 7,79i \end{bmatrix}$$

Setelah didapatkan matriks \hat{R} , selanjutnya dihitung sudut fase sebagai representasi pergeseran objek pada dua citra.

Berdasarkan definisi bahwa $e^{i\theta} = x + iy$ maka

$$e^{i\theta(1,1)} = -0,47 - 9,51i$$

$$\tan \theta = \frac{y}{x} = -\frac{9,51}{-0,47} = 20,23$$

$$\theta = \arctan(20,23)$$

$$\theta = 87^\circ - 180^\circ = -93^\circ = -0,52\pi$$

$$e^{i\theta(1,0)} = 1 - 5,18i$$

$$\tan \theta = \frac{y}{x} = -\frac{5,18}{1} = -5,18$$

$$\theta = \arctan(-5,18)$$

$$\theta = -79^\circ = -0,43\pi$$

Dari sudut fase θ inilah perubahan piksel dapat diketahui. Misalkan pergeseran dua citra dinyatakan sebagai (u,v) dengan nilai pergeseran sebesar (0,1). Artinya *template* pada citra mengalami pergeseran sejauh 0 pada sumbu x dan 1 pada sumbu y. Untuk mendapatkan posisi objek maka akan dihitung :

$$\overline{f_{u,v}} = \frac{1}{N_x N_y} \sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} f(x, y)$$

u, v merupakan pergeseran posisi *template* pada piksel-piksel citra asli. Maka misal pada citra f yang berukuran $M_x \times M_y$ dan matriks citra *template* berukuran $N_x \times N_y$ akan diperoleh perhitungan $\overline{f_{u,v}}$ sebagai berikut :

matriks $f = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$ dan matriks $t = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$ dengan nilai

pergeseran (u, v) sebesar $(0, 1)$ maka akan diperoleh :

$$\begin{aligned} \overline{f_{0,1}} &= \frac{1}{2 \cdot 2} \sum_{x=0}^{0+2-1} \sum_{y=1}^{1+2-1} f(x, y) \\ &= \frac{1}{4} [f(0,1) + f(0,2) + f(1,1) + f(1,2)] \\ &= \frac{1}{4} [2 + 0 + 1 + 1] = \frac{4}{4} = 1 \end{aligned}$$

$$\begin{aligned} \overline{t_{0,1}} &= \frac{1}{1 \cdot 1} \sum_{x=0}^{0+1-1} \sum_{y=1}^{1+1-1} t(x, y) \\ &= [t(0,1)] = [2] = 2 \end{aligned}$$

Diasumsikan bahwa nilai tengah dari *template* t berupa matriks $t' = [1]$, sehingga diperoleh $N'_x \times N'_y$ adalah ukuran matriks \bar{t} yaitu 1×1

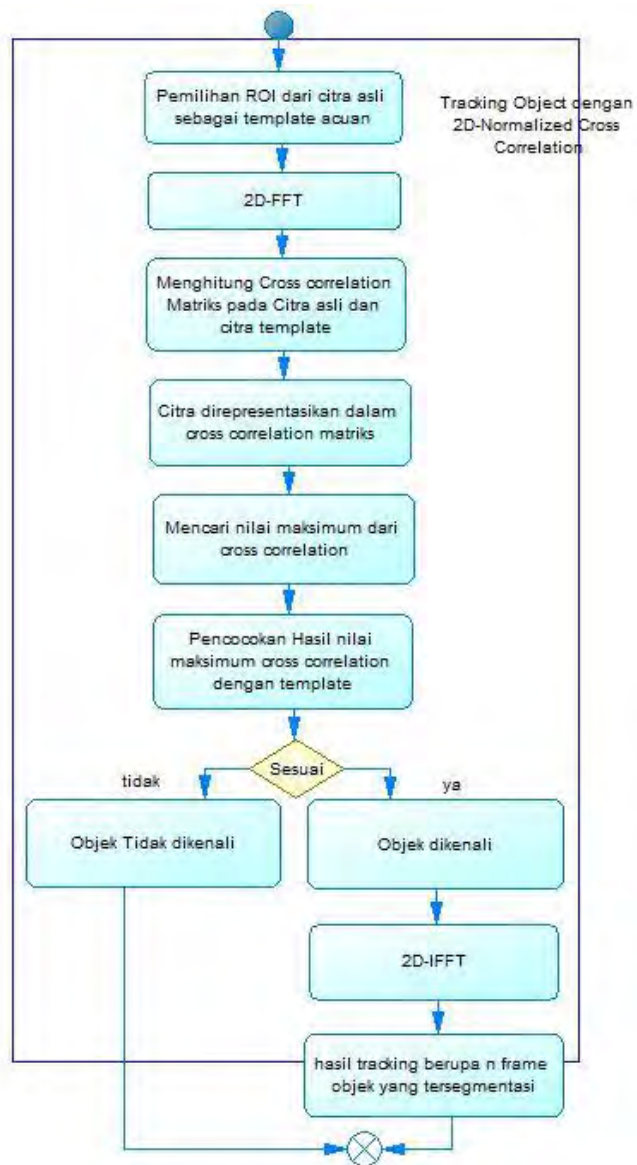
Proses ini berlanjut sebanyak *frame* video yang mengalami pergeseran pada tiap *frame* dari *frame* sebelumnya.

Selanjutnya perhitungan nilai *cross correlation*, γ , menggunakan persamaan (4.1) sehingga diperoleh :

$$\begin{aligned} \gamma &= \frac{\sum_{x=0}^{M_x-1} \sum_{y=0}^{M_y-1} (f(x, y) - 1) (t(x - 0, y - 1) - \bar{t})}{\sqrt{\sum_{x=0}^{M_x-1} \sum_{y=0}^{M_y-1} (f(x, y) - 1)^2 (t(x - 0, y - 1) - \bar{t})^2}} \\ \gamma &= \frac{\sum_{x=0}^2 \sum_{y=0}^2 (f(x, y) - 1) (t(x, y - 1) - 2)}{\sqrt{\sum_{x=0}^2 \sum_{y=0}^2 (f(x, y) - 1)^2 (t(x, y - 1) - 2)^2}} \end{aligned}$$

$$\begin{aligned}
\gamma &= \frac{(f(0,0) - 1)(t(0, -1) - 2)}{\sqrt{(f(0,0) - 1)^2(t(0, -1) - 2)^2}} \\
&\quad + \frac{(f(0,1) - 1)(t(0,0) - 2)}{\sqrt{(f(0,1) - 1)^2(t(0,0) - 2)^2}} \\
&\quad + \frac{(f(0,2) - 1)(t(0,1) - 2)}{\sqrt{(f(0,2) - 1)^2(t(0,1) - 2)^2}} \\
&\quad + \frac{(f(1,0) - 1)(t(1, -1) - 2)}{\sqrt{(f(1,0) - 1)^2(t(1, -1) - 2)^2}} \\
&\quad + \frac{(f(1,1) - 1)(t(1,0) - 2)}{\sqrt{(f(1,1) - 1)^2(t(1,0) - 2)^2}} \\
&\quad + \frac{(f(1,2) - 1)(t(1,1) - 2)}{\sqrt{(f(1,2) - 1)^2(t(1,1) - 2)^2}} \\
&\quad + \frac{(f(2,0) - 1)(t(2, -1) - 2)}{\sqrt{(f(2,0) - 1)^2(t(2, -1) - 2)^2}} \\
&\quad + \frac{(f(2,1) - 1)(t(2,0) - 2)}{\sqrt{(f(2,1) - 1)^2(t(2,0) - 2)^2}} \\
&\quad + \frac{(f(2,2) - 1)t(2,1) - 2}{\sqrt{(f(2,2) - 1)^2(t(2,1) - 2)^2}} \\
\gamma &= 0 + \frac{(1)(-1)}{\sqrt{(1^2)(-1)^2}} + 0 + 0 + 0 + 0 + 0 + 0 + 0 = -1
\end{aligned}$$

Sehingga diperoleh bahwa nilai cross correlation sebesar -1.



Gambar 4.6. Diagram Alir Proses *Tracking* dengan 2D-NCC

4.2.3.2 Perancangan Proses Registrasi Citra

Citra yang telah tesimpan akan diambil sebagai masukan dalam proses registrasi ini. PBIM mampu mengestimasi pergerakan translasi dari dua buah citra yang dibandingkan. Citra $f(n_1, n_2)$ sebagai citra ke- i dan citra $g(n_1, n_2)$ sebagai citra ke- $i+1$ pada serangkaian citra registrasi yang dibandingkan dan dikenai 2D discrete fourier transform (2D-DFT).

Hasil dari proses DFT ini akan dimanfaatkan untuk mendapatkan nilai dari cross spektrum menggunakan persamaan :

$$\hat{R}(k_1, k_2) = \frac{F(k_1, k_2)\overline{G(k_1, k_2)}}{|F(k_1, k_2)\overline{G(k_1, k_2)}|} = e^{j\theta(k_1, k_2)}$$

Andaikan $f_c(x_1, x_2)$ sebagai suatu nilai piksel dari citra dua dimensi dengan index posisi x_1 dan x_2 . Diberikan δ_1 dan δ_2 merepresentasikan perpindahan sub piksel dari $f_c(x_1, x_2)$ pada x_1 dan x_2 sehingga diperoleh bahwa citra dengan perpindahan tersebut adalah $f_c(x_1 - \delta_1, x_2 - \delta_2)$. Asumsikan bahwa $f(n_1, n_2)$ dan $g(n_1, n_2)$ adalah titik sampel spasial dari $f_c(x_1, x_2)$ dan $f_c(x_1 - \delta_1, x_2 - \delta_2)$ sehingga didefinisikan sebagai :

$$\begin{aligned} f(n_1, n_2) &= f_c(x_1, x_2) | x_1 = n_1 T_1, x_2 = n_2 T_2 \\ g(n_1, n_2) &= f_c(x_1 - \delta_1, x_2 - \delta_2) | x_1 = n_1 T_1, x_2 \\ &= n_2 T_2 \end{aligned}$$

Dengan T_1, T_2 adalah sampel spasial dari interval dan $n_1 = -M_1, \dots, M_1$, $n_2 = -M_2, \dots, M_2$. Sehingga diperoleh $F(k_1, k_2)$ dan $G(k_1, k_2)$ sebagai transformasi fourier diskrit dari $f(n_1, n_2)$ dan $g(n_1, n_2)$. Berdasarkan perbedaan domain pada transformasi fourier yang didefinisikan dalam domain frekuensi dan citra asli dalam domain spasial, maka :

$$G(k_1, k_2) \cong F(k_1, k_2) \cdot e^{-\frac{j2\pi}{N_1}k_1\delta_1} e^{-\frac{j2\pi}{N_2}k_2\delta_2}$$

Sehingga diperoleh bahwa :

$$\begin{aligned}\hat{R}(k_1, k_2) &\cong e^{-\frac{j2\pi}{N_1}k_1\delta_1} e^{-\frac{j2\pi}{N_2}k_2\delta_2} \\ e^{j\theta(k_1, k_2)} &\cong e^{-\frac{j2\pi}{N_1}k_1\delta_1} e^{-\frac{j2\pi}{N_2}k_2\delta_2} \\ \theta(k_1, k_2) &= \frac{2\pi k_1}{N_1}\delta_1 + \frac{2\pi k_2}{N_2}\delta_2\end{aligned}\quad (4.2)$$

Dari perhitungan sudut fase pada subbab 4.2.3.1 yang menghasilkan $\theta_1 = -0,52\pi$ dan $\theta_2 = -0,43\pi$.

Dengan menggunakan Persamaan (4.2)

$$\begin{aligned}\theta(1,1) &= \frac{2\pi}{3}\delta_1 + \frac{2\pi}{3}\delta_2 \\ -0,52\pi &= \frac{2\pi}{3}(\delta_1 + \delta_2) \\ -\frac{1,56}{2} &= \delta_1 + \delta_2 \\ -0,78 &= \delta_1 + \delta_2\end{aligned}\quad (4.3)$$

$$\begin{aligned}\theta(1,0) &= \frac{2\pi}{3}\delta_1 + \frac{2\pi}{3} \cdot 0 \cdot \delta_2 \\ -0,43\pi &= \frac{2\pi}{3}\delta_1 \\ -0,64 &= \delta_1\end{aligned}\quad (4.4)$$

Dengan mensubstitusikan Persamaan (4.4) pada Persamaan (4.3) maka akan didapatkan:

$$\delta_2 = -0,78 + 0,64 = -0,14$$

Sehingga diperoleh $\delta_1 = -0,64$ dan $\delta_2 = -0,14$.

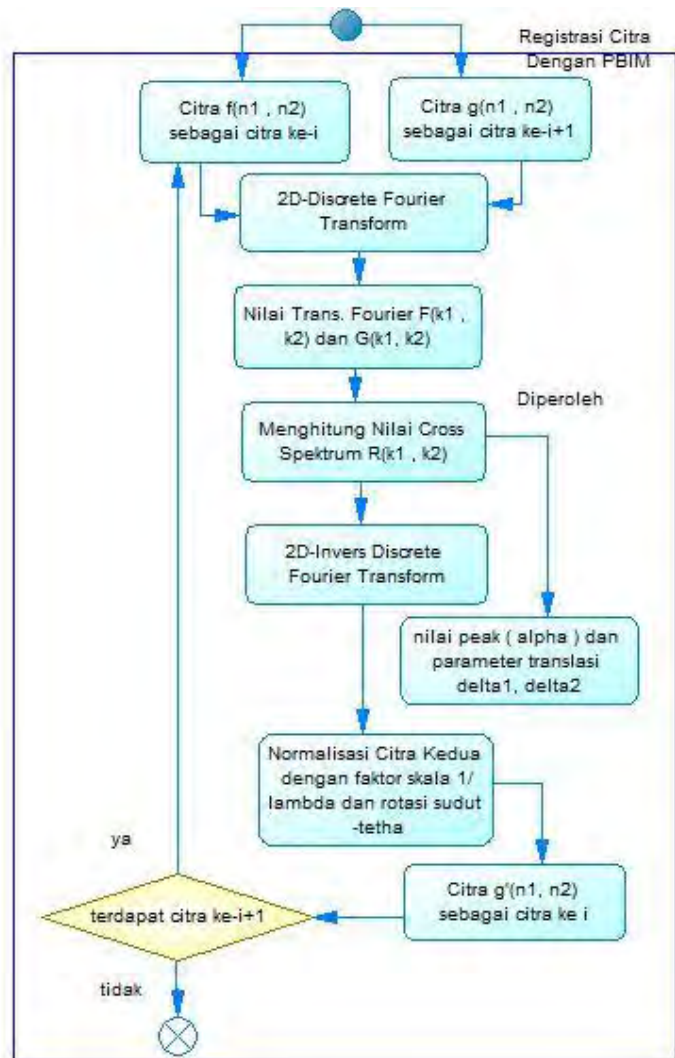
Selanjutnya hasil dari citra yang direpresentasikan dalam nilai cross spektrum dikonversi kembali dengan invers transformasi fourier diskrit.

Contoh perhitungan invers Tranformasi Fourier :

$$\begin{aligned}
 f(n_1, n_2) &= \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} F(k_1, k_2) e^{2\pi i (\frac{k_1 n_1}{N_1} + \frac{k_2 n_2}{N_2})} \\
 f(0,0) &= \frac{1}{3 \cdot 3} \sum_{n_1=0}^2 \sum_{n_2=0}^2 F(k_1, k_2) \\
 &= \frac{1}{9} [F(0,0) + F(0,1) + F(0,2) + F(1,0) + F(1,1) \\
 &\quad + F(1,2) + F(2,0) + F(2,1) + F(2,2)] \\
 &= \frac{1}{9} [8 + 0,5 - 0,86i + 0,5 - 0,86i + 0,5 - 2,59i + \\
 &\quad (-1) + 3,46i + 0,5 + 0,86i + 0,5 + 2,59i + 0,5 - \\
 &\quad 0,86i + (-1) + 3,46i] = \frac{9}{9} = 1
 \end{aligned}$$

Dengan cara yang sama maka akan diperoleh elemen matriks yang lain. Pemanfaatan Invers DFT ini menghasilkan nilai peak, α , yang merepresentasikan tingkat perubahan posisi dari dua buah citra. Misal pada citra identik dan citra yang bergeser.

Nilai *displacement*, (δ_1, δ_2) disimpan sebagai parameter translasi objek dalam citra sejauh δ_1 dan δ_2 pada sumbu x dan y. Normalisasi citra dilakukan dengan faktor skala $\frac{1}{\lambda}$ dan sudut rotasi $-\theta$ agar citra yang ternormalisasi dapat dibandingkan kembali dengan citra yang lain untuk mendapatkan skala parameter yang lain. Proses ini berulang sebanyak n *frame* citra observasi yang dibutuhkan.



Gambar 4.7. Diagram Alir Proses Registrasi Citra dengan PBIM

4.2.3.3 Perancangan Proses Rekonstruksi Citra

Rekonstruksi citra dilakukan dengan memproyeksikan nilai translasi terhadap operator proyeksi kepada himpunan convex tertutup menggunakan persamaan:

$$x^{(n+1)} = P_m P_{m-1} \dots P_2 P_1 x^{(n)}$$

Selanjutnya dilakukan perhitungan untuk mendapatkan nilai x^0 yang merupakan perbesaran citra dengan algoritma interpolasi. Dimana operator proyeksi dari tiap piksel $x(n_1, n_2, k)$ terhadap $C_{m_1, m_2, l}$ didefinisikan sebagai :

$$P_{m_1, m_2, l}[x(n_1, n_2, k)] = \begin{cases} x + \frac{r^{(x)} - \delta_0}{H} h, & r^{(x)} > \delta_0 \\ x, & -\delta_0 \leq r^{(x)} \leq \delta_0 \\ x + \frac{r^{(x)} + \delta_0}{H} h, & r^{(x)} < -\delta_0 \end{cases}$$

dan penghitungan nilai piksel dilakukan dengan persamaan :

$$r^{(x)}(m_1, m_2, l) = g(m_1, m_2, l) - \sum_{n_1, n_2} f(n_1, n_2, k) \cdot h(m_1, m_2; n_1, n_2, l)$$

$h(m_1, m_2; n_1, n_2, l)$ adalah pergeseran linear dari *point spread function* untuk setiap indeks posisi dalam *frame* hasil perbesaran interpolasi. Iterasi dilakukan untuk mendapatkan setiap nilai estimasi piksel dalam *frame* hasil interpolasi sampai kriteria pemberhentian diperoleh. Setelah iterasi berhenti, matriks citra akan diproyeksikan kembali dengan operator proyeksi P dan diperoleh nilai-nilai piksel dari citra baru yang disebut dengan citra super resolusi.

Contoh perhitungan POCS :

Diberikan point spread function, $h(m_1, m_2; n_1, n_2, l) = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$, $\delta_0 = 2$, $f(n_1, n_2, k)$ merupakan citra super resolusi yang akan dibentuk, dimana $f(n_1, n_2, k)$ mengalami penambahan pergeseran citra hasil registrasi atau dapat dinyatakan dengan $f(m_1 + \delta_a, m_2 + \delta_b)$ a merepresentasikan sumbu x dan b merepresentasikan sumbu y. Diasumsikan bahwa $\delta_x = 0$ dan $\delta_y = 1$.

Citra resolusi rendah $g(x, y) = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$

Misal akan dicari nilai dari

$$\begin{aligned} r^{(1)}(1,1,1) &= 1 - [f(0,0)h(1,1; 0,0, 1) + f(0,1)h(1,1; 0,1, 1) \\ &\quad + f(0,2)h(1,1; 0,2, 1) + f(1,0)h(1,1; 1,0, 1) \\ &\quad + f(1,1)h(1,1; 1,1, 1) + f(1,2)h(1,1; 1,2, 1)] \\ &= r_1 \end{aligned}$$

Maka r_1 akan dikalikan dengan operator proyeksi sampai memenuhi batas δ_0 . Dengan cara yang sama r pada tiap piksel akan dihitung untuk memperoleh nilai piksel citra super resolusi.



Gambar 4.8. Diagram Alir Proses Rekonstruksi Citra dengan POCS

4.2.4 Perancangan Data

Terdapat tiga macam data yang digunakan oleh sistem ini antara lain data masukan, data proses, dan data keluaran. Pada Tugas Akhir ini data masukan berupa video rekaman *offline* yang diambil oleh peneliti dan video sampel objek bergerak. Data proses merupakan data yang berisi parameter-parameter yang akan digunakan oleh algoritma *2D Normalized Cross Correlation* (2D-NCC) dalam proses tracking, metode *Phase based image matching* (PBIM) dan *Projection onto convex sets algorithm* (POCS). Sedangkan data keluaran adalah data citra hasil super resolusi.

4.2.4.1 Data Masukan

Data masukan sistem ini berupa video rekaman *offline* yang diambil oleh peneliti dan video sampel objek bergerak. File video yang diambil memiliki spesifikasi yaitu:

- a. Memiliki kemampuan untuk merekam video dengan kecepatan minimal *25 frame per second*.
- b. Memiliki kemampuan untuk merekam video dengan resolusi minimal 320 x 240 piksel citra.
- c. Format ekstensi rekaman video *offline* dapat berupa: *.*avi*.

4.2.4.2 Data Proses

Data proses merupakan data yang digunakan dalam proses pengolahan data masukan. Data proses ini diperoleh dari hasil pengolahan data masukan sesuai dengan tahapan algoritma dan metode yang telah disusun. Tabel 4.2 menjelaskan tahapan dari data proses.

Tabel 4.2. Tabel data proses

No	Tahapan	Input	Output
1.	<i>Input Awal</i>	Video	<i>Frame Citra</i>
2.	Pilih Area ROI	<i>Frame Citra</i>	Citra ROI
3.	Segmentasi citra ROI	Citra ROI	Template Citra untuk <i>tracking</i>
4.	Proses Tracking dengan <i>2D-NCC</i>	Template Citra untuk <i>tracking</i>	Bounding Box pada objek tracking
5.	Segmentasi citra pada bounding box	Bounding Box pada objek tracking	Citra hasil segmentasi yang disimpan dalam <i>directory</i>
6.	Proses Registrasi Super resolusi dengan metode PBIM	Citra hasil segmentasi yang disimpan dalam <i>directory</i>	Parameter rekonstruksi
7.	Proses rekonstruksi Super Resolusi dengan metode POCS	Parameter rekonstruksi	Citra Super Resolusi
8.	Proses pengujian hasil Super resolusi	Citra Super Resolusi	Nilai PSNR

4.2.4.3 Data Keluaran

Data keluaran berupa citra hasil Super Resolusi dan nilai PSNR. Program ini juga menghasilkan keluaran berupa citra ROI, Template Citra untuk *tracking*, Citra hasil segmentasi pada bounding box yang disimpan dalam *directory*, dan Parameter rekonstruksi.

4.2.5 Perancangan Antar Muka Sistem

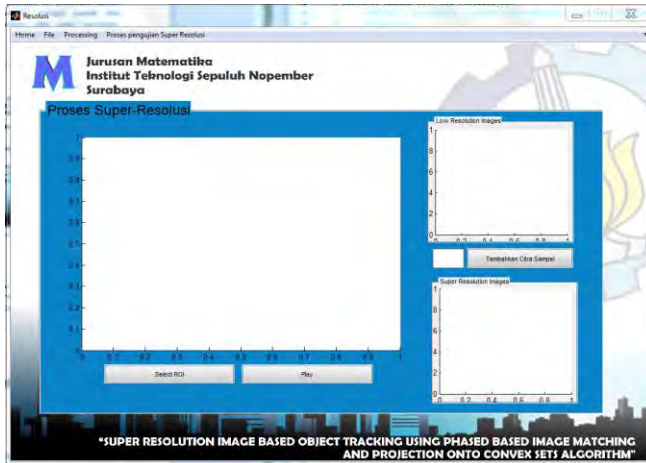
Desain antar muka sistem dibutuhkan agar pengguna dengan mudah mengoperasikan perangkat lunak yang dibangun.

4.2.5.1 Perancangan Halaman Utama

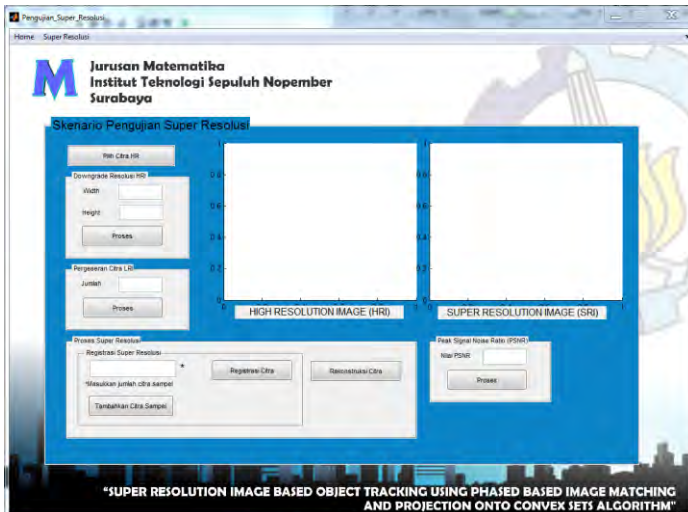
Halaman utama ini memiliki desain antar muka yang ditunjukkan pada Gambar 4.10. Pada halaman ini dilakukan proses pemilihan video, pemilihan ROI, *tracking*, mengisi jumlah citra sampel, menampilkan citra *low resolution*, citra *super resolution*, citra *high resolution*, dan nilai PSNR.



Gambar 4.9. Antar muka halaman awal.



Gambar 4.10. Antar muka halaman proses *tracking* dan super resolusi.



Gambar 4.11. Antar muka halaman proses pengujian super resolusi.

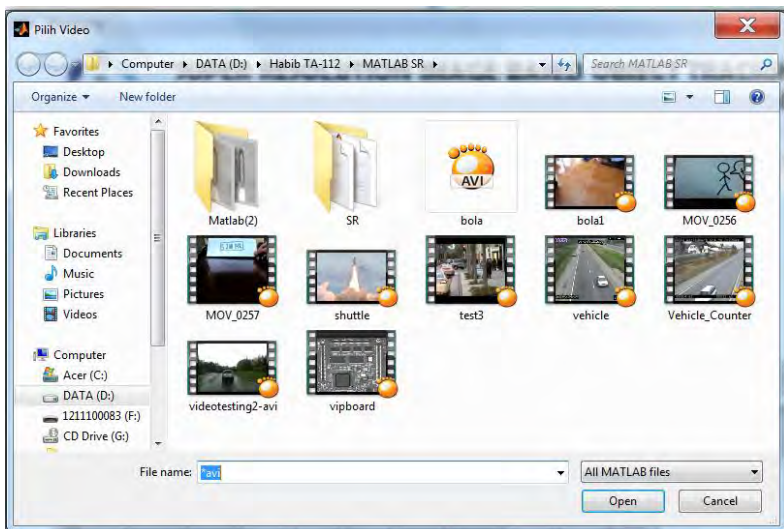
4.3 Implementasi Sistem

4.3.1 Implementasi *Input* Video

Inputan program berupa video *single objek* dengan format avi. Dianjurkan untuk menggunakan video yang memiliki kecepatan minimal 25 fps, supaya proses pengolahan informasi digital berlangsung dengan baik. Proses tersebut diimplementasikan dalam program berikut :

```
[filename, pathname] = uigetfile({'*avi'},
    'Pilih Video'); % akses directory file video
vid=strcat(pathname, filename);
videoReader = vision.VideoFileReader(vid,
    'ImageColorSpace', 'Intensity')
img = step(videoReader);
axes(handles.axes1);
imshow(img);
```

Kode program lebih lengkap dapat dilihat pada lampiran A.. Gambar 4.12 adalah tampilan antar muka pengambilan *input* video.



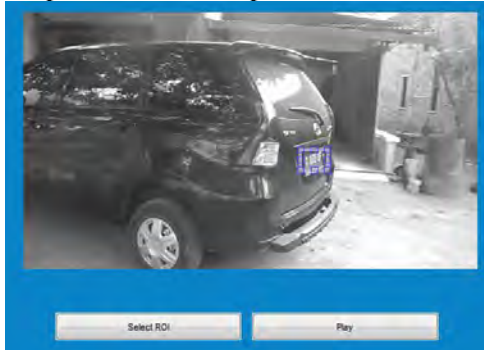
Gambar 4.12. Antar muka *input* video

4.3.2 Implementasi Pemilihan area ROI

Region of Interest (ROI) adalah daerah bagian dari citra atau *frame* yang akan diproses dalam sistem. Pemilihan ROI ini bertujuan untuk memudahkan dan membantu program dalam menspesifikasikan objek yang dibutuhkan. Ukuran ROI pada sistem ini disesuaikan dengan ukuran objek yang dibutuhkan, sehingga piksel-piksel yang berada diluar ROI tidak dianggap sebagai objek dan tidak diproses. Hasil pemilihan ROI ini berupa nilai untuk menentukan centroid dari objek dan berupa segmentasi objek tiap *frame* yang disimpan dalam direktori. Proses tersebut diimplementasikan dalam program berikut :

```
h = imrect;
posisi = wait(h);
get_roi = getPosition(h)
centroid = [get_roi(1)+uint32(get_roi(3)/2)
get_roi(2)+uint32(get_roi(4)/2)]
target_img = imcrop(img,get_roi);
axes(handles.axes4);
imshow(target_img);
```

Kode program selengkapnya disajikan pada Lampiran A. Gambar 4.13 adalah tampilan antar muka pemilihan area ROI.



Gambar 4.13. Antar muka pemilihan area ROI dengan Kotak warna biru pada *frame* tersebut adalah area ROI yang dipilih.

4.3.3 Implementasi Proses *Tracking* dengan 2D-Normalized Cross Correlation

Proses ini merupakan pendekatan standar dalam deteksi ciri (feature detection). Dimana jika diberikan dua citra, yaitu f sebagai citra uji dan t sebagai citra *template*. Jika $f(x,y)$ menotasikan nilai intensitas dari citra f yang berukuran $M_x \times M_y$ pada titik (x,y) , $x,y \in \{0, \dots, M_{x,y} - 1\}$. Sedangkan *template* t yang diberikan berukuran $N_x \times N_y$. Selanjutnya untuk menentukan posisi hasil pencarian adalah dengan melakukan penghitungan nilai *normalized cross correlation*, γ , pada tiap titik (u,v) untuk citra f dan *template* t menggunakan persamaan :

$$\gamma = \frac{\sum_{x,y} (f(x,y) - \bar{f}_{u,v})(t(x-u, y-v) - \bar{t})}{\sqrt{\sum_{x,y} (f(x,y) - \bar{f}_{u,v})^2 \sum_{x,y} (t(x-u, y-v) - \bar{t})^2}}$$

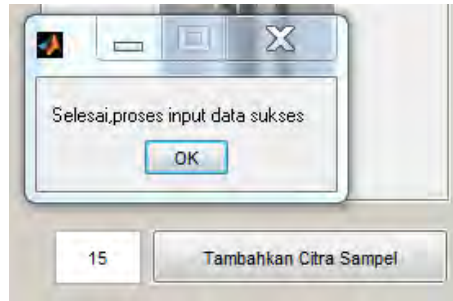
Berikut kode dari proses ini:

```
% menghitung normalized cross correlation.
norm_Corr_f = (corrOutput_f) ./ (IUT *
target_energy);
xyLocation = step(hFindMax, norm_Corr_f);
% menghitung indeks linier
linear_index = sub2ind([ri-rt, ci-ct]+1,
xyLocation(:,2), ...
xyLocation(:,1));
norm_Corr_f_linear = norm_Corr_f(:);
norm_Corr_value =
norm_Corr_f_linear(linear_index);
detect = (norm_Corr_value > threshold);
target_roi = zeros(length(detect), 4);
ul_corner = (gain.*(xyLocation(detect, :)-
1))+1;
target_roi(detect, :) = [ul_corner,
fliplr(target_size(detect, :))];
```

Kode program selengkapnya disajikan pada Lampiran B.

4.3.4 Implementasi Proses Penginputan Citra Observasi untuk Registrasi Super Resolusi

Proses ini dilakukan untuk memasukkan sejumlah data citra objek yang telah disimpan dalam direktori sebagai data sampel registrasi super resolusi menggunakan metode PBIM.



Gambar 4.14. Tampilan untuk memasukkan data citra sampel

Berikut kode dari proses ini:

```
n = get(handles.jml, 'String');
n = str2num(n);
for i = 1:n
    im{i} = imread(['D:\Habib TA-112\MATLAB
SR\Matlab(2)\image' num2str(i) '.jpg']);
    s{i} = imread(['D:\Habib TA-112\MATLAB
SR\Matlab(2)\image' num2str(i) '.jpg']);
end
msgbox('Selesai, proses input data sukses')
```

Kode lebih lengkap pada Lampiran A.

4.3.5 Implementasi Registrasi Citra Menggunakan Metode *Phased based Image Matching* (PBIM)

Metode *Phased based Image Matching* (PBIM) merupakan metode yang dapat digunakan untuk mengestimasi nilai perpindahan posisi (*displacement*) objek pada citra dengan akurasi sub piksel dari posisi puncak korelasi. Selain itu, sudut

rotasi citra dan faktor skala dari pengurangan atau perbesaran citra dapat diketahui dengan mengkonversikannya menjadi nilai translasi citra. Oleh karena itu estimasi translasi perpindahan citra dengan akurasi yang tinggi sangat dibutuhkan [14].

```
delta_est =
      0      0
  1.0340 -0.7690
  2.0080  0.0560
  2.5980 -1.2180
  2.7440 -2.9610
 -0.3880 -0.5820
  0.4160 -3.5290
  1.9170 -0.0580
 -0.1770 -2.5210
  1.2270 -0.3050
  2.8410 -2.6820
  0.1090 -0.4130
  0.8160 -3.4100
  2.2840 -1.4580
 -0.1900 -2.4970
  0.4690 -1.9030
  2.2220 -1.8500
  0.1100  0.3970
  1.5670 -0.6740
 -0.1140 -1.6500
```

Gambar 4.15. delta_est merupakan nilai perpindahan (*displacement*) objek pada kordinat (x,y) untuk 20 citra sampel.


Kode program untuk Registrasi citra adalah sebagai berikut:

```
function [delta_est, phi_est] = PBIM(im)
```

Kode selengkapnya pada Lampiran C.

4.3.6 Implementasi Rekonstruksi Citra Menggunakan Metode *Projection Onto Convex Sets Algorithm (POCS)*

Nilai parameter translasi dari proses registrasi diimplementasikan dalam metode *Projection Onto Convex Sets (POCS)* untuk merekonstruksi dan meningkatkan resolusi citra. Metode ini digunakan untuk mengestimasi citra super resolusi dari rangkaian citra yang dibatasi dalam himpunan convex tertutup dan memperoleh hasilnya melalui proses iterasi[8].

	-Hasil Rekonstruksi dari objek video shuttle.avi -Berdimensi 160x112, dibangun dari citra dengan dimensi 79x61 dengan citra observasi berjumlah 20 citra
	-Hasil Rekonstruksi dari objek video pesawat.avi -Berdimensi 640x204, dibangun dari citra dengan dimensi 315x102 dengan citra observasi berjumlah 20 citra

Gambar. 4.16 Citra Super Resolusi

Kode program untuk Rekonstruksi citra adalah sebagai berikut:

```
function y = pocs(s,delta_est,factor)
```

Kode selengkapnya pada Lampiran D.

BAB V



PENGUJIAN DAN PEMBAHASAN

Pada bab ini akan dilakukan pengujian serta pembahasan terhadap program yang telah dibuat. Pengujian sistem ini merupakan pengujian super resolusi berdasarkan data video yang tersedia dalam penyimpanan (direktori) komputer dan diperoleh dari situs <http://youtube.com>. Pengujian dilakukan pada video pesawat.avi dan car.avi untuk mendapatkan nilai PSNR dan waktu komputasi.

5.1 Data Uji Coba

Uji coba pada program dalam penelitian tugas akhir ini dilakukan terhadap video berekstensi .avi sebanyak tiga video.

Tabel 5.1 Data Video yang Digunakan.

No	Nama	Jumlah Frame	Video
1	car.avi	29	
2	pesawat.avi	150	

5.2 Pengujian Tahap Pemilihan ROI

Proses ini bertujuan untuk mengetahui bahwa prototipe perangkat lunak berhasil memperoleh citra objek yang akan dilacak sebagai *template*. Gambar dan ukuran dari setiap citra objek yang akan dilacak dapat dilihat pada Gambar 5.1.



Gambar 5.1. Citra objek hasil segmentasi dengan ROI

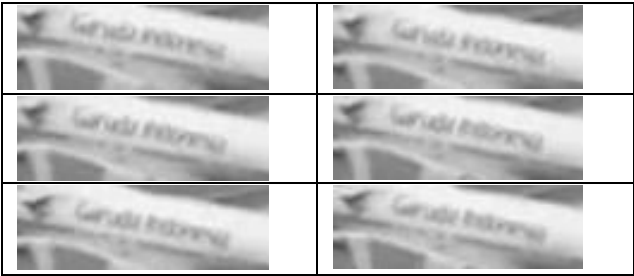
5.3 Pengujian Hasil Super Resolusi

Untuk pengujian hasil dari super resolusi ini akan diambil n citra dari hasil objek *tracking* sebagai citra masukan untuk proses registrasi menggunakan metode *phased based image matching* (PBIM) dan selanjutnya akan dilakukan proses rekognstruksi menggunakan metode *projection onto convex sets algorithm* (POCS). Setelah itu citra hasil rekonstruksi akan dibandingkan dengan citra referensinya untuk mendapatkan nilai PSNR.

5.3.1 Pengujian PSNR (*Peak Signal to Noise Ratio*)

Pengujian PSNR ini dilakukan pada video pesawat.avi, yaitu dengan pengambilan area ROI pada badan pesawat yang bertuliskan “Garuda Indonesia”. Berikut hasil pengujian tersebut :

- Nama Video : pesawat.avi
- Jumlah frame : 150
- Ukuran citra Observasi : 112x48 piksel
- Ukuran citra hasil Rekonstruksi : 224x96 piksel



Gambar 5.2 Rangkaian Citra Hasil Tracking Resolusi Rendah pada ROI badan pesawat



(a)



(b)

Gambar 5.3. Super resolusi badan pesawat Garuda Indonesia (a) Citra hasil rekonstruksi (b) citra asli dikenai perbesaran interpolasi 2 kali

Tabel 5.2 Hasil pengujian PSNR pada video pesawat.avi

Ukuran	Jumlah Observasi	PSNR (dB)
Segmentasi: 112x48 piksel	10	29,586
	20	29,586
	30	30,5413
	40	30,5413
	50	30,5413
Rata-rata		30,159

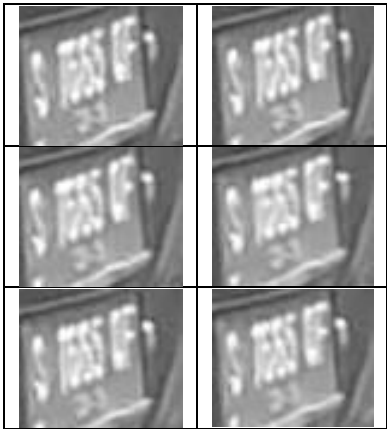
Pengujian kedua dilakukan pada video car.avi dengan hasil sebagai berikut :

Nama Video : car.avi

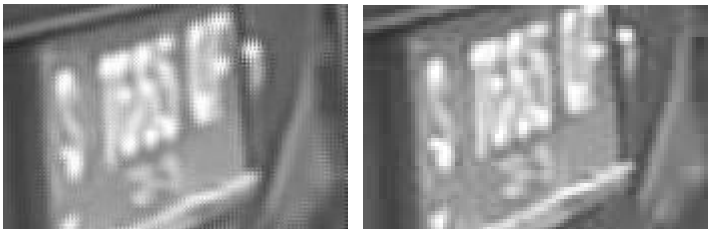
Jumlah frame : 29

Ukuran citra Observasi : 86x56 piksel

Ukuran citra hasil Rekonstruksi : 172x112 piksel



Gambar 5.4. Rangkaian citra hasil tracking resolusi rendah car.avi



(a) (b)

Gambar 5.5. Super resolusi plat nomor mobil (a) hasil Super resolusi 172x112 piksel (b) citra asli dengan perbesaran interpolasi 2 kali

Tabel 5.3 Hasil pengujian pada video car.avi dengan segmentasi ROI 86x56 piksel

Ukuran	Jumlah Observasi	PSNR
Segmentasi: 86x56 piksel	10	29,3694
	20	29, 6905
	30	29, 6906
	40	30,3183
	50	30,3183
Rata-rata		30,3183

5.3.2 Pengujian Waktu Komputasi

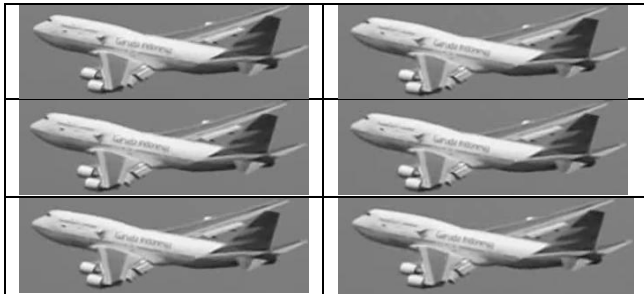
Pada pengujian pertama dilakukan pada video pesawat.avi akan diambil ROI pada keseluruhan objek pesawat dan pada tulisan “Garuda Indonesia”. Hal ini dilakukan untuk mendapatkan informasi tentang efektivitas waktu komputasi dari proses super resolusi yang dilakukan. Berikut hasil dari pengujian tersebut :

Nama Video : pesawat.avi

Jumlah frame : 150

Ukuran citra observasi pada seluruh bagian objek pesawat :
324x118 piksel

Ukuran citra hasil rekonstruksi pada seluruh bagian objek pesawat: 648x236 piksel



Gambar 5.6 Rangkaian Citra Hasil Tracking Resolusi Rendah pesawat



(a)



(b)

Gambar 5.7. Hasil Super resolusi objek pesawat (a) Citra hasil Rekonstruksi (b) Citra Asli yang dikenai perbesaran interpolasi dua kali

Tabel 5.4 Waktu komputasi pada video pesawat.avi

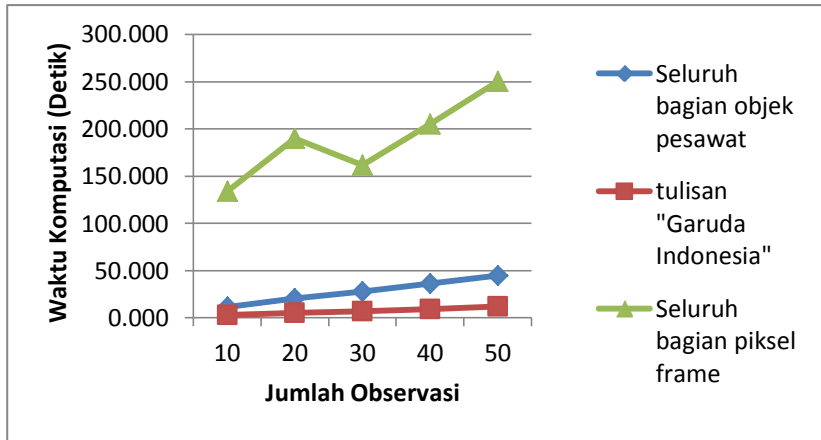
Ukuran	Jumlah Observasi	Waktu Komputasi (detik)
Segmentasi: 324x118 piksel (pada keseluruhan bagian objek pesawat)	10	11,3205
	20	20,3996
	30	27,7954
	40	36,3199
	50	44,6954
Rata-rata		28,1061
Segmentasi: 112x48 piksel (pada tulisan “Garuda Indonesia”)	10	3,2259
	20	5,50233
	30	7,119
	40	9,360
	50	12,073
Rata-rata		7,4560
Seluruh bagian piksel <i>frame</i>	10	134,281
	20	190,162
	30	161,475
	40	205,435
	50	250,181
Rata-rata		188,306

Pengujian waktu komputasi pada video car.avi dengan hasil pengujian sebagai berikut :

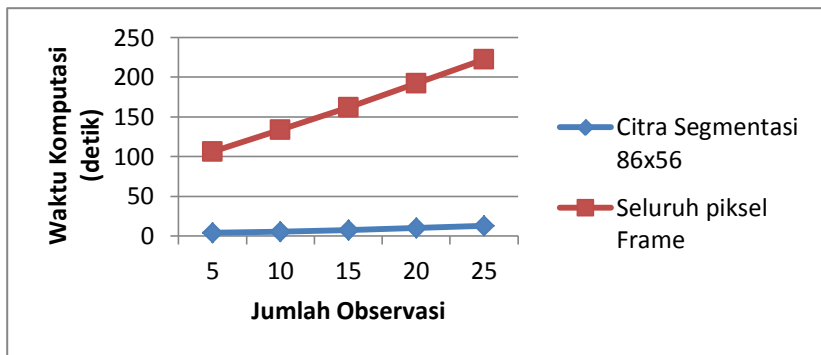
Tabel 5.5 Hasil Pengujian waktu komputasi pada video car.avi

Ukuran	Jumlah Observasi	Waktu Komputasi (dalam detik)
Segmentasi: 86x56 piksel	5	3,75279
	10	5,38669
	15	7,5329
	20	10,0362
	25	12,4993
Rata-rata		7,84158
Keseluruhan <i>Frame</i>	5	105,99
	10	133,479
	15	162,203
	20	192,442
	25	222,278
Rata-rata		163,278

Berdasarkan hasil percobaan pada tabel 5.2 dan 5.3 dapat diperoleh bahwa nilai PSNR terbaik berbanding lurus dengan jumlah citra observasi, dimana semakin banyak citra observasi maka nilai PSNR akan meningkat. Hal ini disebabkan oleh ketersediaan informasi tiap piksel yang diperoleh dalam proses super resolusi lebih banyak. Selain itu, perbandingan waktu komputasi pada citra *tracking* dan seluruh piksel *frame* dalam proses super resolusi juga terjadi perbedaan yang signifikan. Hasil perbandingan tersebut disajikan dalam grafik berikut :



Gambar 5.8 Grafik perbandingan waktu komputasi pada pengujian video pesawat.avi.



Gambar 5.8 Grafik perbandingan waktu komputasi pada pengujian video car.avi.

Berdasarkan data grafik diatas menunjukkan bahwa citra tracking memiliki waktu komputasi yang relatif lebih singkat dalam proses super resolusi dibandingkan dengan waktu komputasi pada seluruh piksel *frame*. Hal ini disebabkan karena proses super resolusi pada citra *tracking* dilakukan pada piksel yang lebih sedikit.

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisi tentang beberapa kesimpulan yang dihasilkan berdasarkan penelitian yang telah dilaksanakan. Di samping itu, pada bab ini juga dimasukkan beberapa saran yang dapat digunakan jika penelitian ini ingin dikembangkan.

6.1 Kesimpulan

Berdasarkan analisis terhadap hasil pengujian yang telah dilakukan terhadap sistem super resolusi berbasis citra *tracking* dengan metode *phased based image matching* dan *projection onto convex sets*, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Sistem berhasil melakukan *tracking* pada bagian tertentu dari objek bergerak. Hasil *tracking* ini berupa citra resolusi rendah yang selanjutnya digunakan sebagai masukan dalam proses super resolusi.
2. Sistem berhasil melakukan proses registrasi untuk mendapatkan nilai parameter translasi menggunakan metode *phased based image matching* dan berhasil melakukan proses rekonstruksi citra super resolusi menggunakan metode *projection onto convex set*.
3. Hasil perhitungan PSNR menunjukkan bahwa citra hasil Super Resolusi memberikan kualitas lebih baik jika jumlah citra pada citra observasi semakin banyak. Nilai PSNR rata-rata untuk video car.avi sebesar 30,3183 dB dan pesawat.avi sebesar 30,159 dB.
4. Hasil perbandingan waktu komputasi pada citra hasil *tracking* dengan keseluruhan piksel *frame* menunjukkan efektivitas citra *tracking* sebagai masukan dalam proses komputasi super resolusi, yaitu misal pada pengujian video pesawat.avi diperoleh waktu komputasi citra *tracking* pada seluruh piksel objek pesawat sebesar 28,1061 detik, citra *tracking* pada tulisan “Garuda Indonesia” sebesar 7,456 detik

dan waktu komputasi untuk seluruh piksel *frame* sebesar 188,306.

6.2 Saran

Berdasarkan hasil yang dicapai pada penelitian ini, ada beberapa hal yang penulis sarankan untuk pengembangan selanjutnya yaitu:

1. Program belum bisa melakukan *tracking* pada *multiple object*, oleh sebab itu untuk penelitian selanjutnya proses *tracking* dilakukan pada *multiple object* sebagai *input* proses super resolusi. Selain itu, perlu dilakukan pemilihan metode *tracking* yang lebih *adaptive* terhadap objek, sehingga proses *tracking* pada objek bergerak dapat lebih akurat dan bervariasi.
2. Hasil dari proses super resolusi pada sistem ini masih berupa citra gambar dua dimensi. Untuk penelitian selanjutnya hasil dari super resolusi berupa *video sequence*.
3. Program belum terhubung dengan database. Sehingga data penghitungan hanya disimpan sementara oleh memori CPU. Pada penelitian berikutnya program dapat dihubungkan dengan database. Sehingga data dapat disimpan dan diolah untuk kepentingan lebih lanjut.

DAFTAR PUSTAKA

- [1] Bryandi, Achmad. (2011). “Rekonstruksi Citra Pada Super Resolusi menggunakan Projection Onto Convex Sets”. **Tugas Akhir. Jurusan Matematika ITS**
- [2] Setiyono, Budi., Hariadi, M.,Purnomo, Heri Mauridhi. (2012). “Superresolution Using Papoulis-Gerchberg Algorithm Based Phase Based Image Matching”. **Jurnal Ilmiah Kursor. Vol.6 no.3**
- [3] Setiyono, Budi, Hariadi, M.,Purnomo, Heri Mauridhi. (2012). “Survey Of Superresolution Using Phased Based Image Matching”. **Journal Of Theoretical And Applied Information Technology. Vol. 43 No. 2**
- [4] Rafael C. Gonzales, Richard E. Woods. (2002). “Digital Image Processing”. **United States of America: Tom Robbins Publisher.**
- [5] Al Bovik. (2000). “Handbook of Image and Video Processing”. **San Diego: Academic Press Publisher.**
- [6] Frederick W. Wheeler, Ralph T. Hocter, Eamon B. Barrett. (2005). ”Super-Resolution Image Synthesis using Projections onto Convex Sets in the Frequency Domain”. **IS&T/SPIE Symposium on Electronic Imaging, Conference on Computational Imaging. Vol. 5674.** San Jose. pp. 479-490, January.
- [7] H. Stark and P. Oskoui, “High resolution image recovery from image plane arrays,using convex projections,” **J. Opt. Soc. Am. A, vol. 6,** pp.1715-1726, 1989.

- [8] Hao,S., Lin, L.,Weiping, Z. Limin, L. (2009). "Location and Super-resolution Enhancement of License Plates Based on Video Sequences". **Laboratory Of Image Science And Technology, College Of Computer Science And Engineering. Southeast University Nanjing, China.**

- [9] Lazuardi, R. Arif Firdaus. (2014). "Penghitungan Kendaraan Bergerak Berbasis Algoritma Background Subtraction Menggunakan Metode Gaussian Mixture Model". **Tugas Akhir. Jurusan Matematika ITS.**

- [10] id.wikipedia.org(2013).”NTMC Polri”.
[http://id.wikipedia.org/wiki/NTMC Polri,](http://id.wikipedia.org/wiki/NTMC_Polri) *posted: 2*
 Maret 2013. Diakses pada tanggal 17-02-2014

- [11] Yilmaz, A., Javed, O., and Shah, M. (2006). "Object tracking: A survey". *ACM Comput. Surv.* **38, 4, Article 13 (Dec. 2006).**

- [12] mathworks.com. (2015). "Spesifying Region Of Interest (ROI)".
[http://www.mathworks.com/help/images/specifying-a-region-of-interest-roi.html.](http://www.mathworks.com/help/images/specifying-a-region-of-interest-roi.html) Diakses pada 18-02-2015

- [13] Santosh, D. Hari Hara et all. (2013). "Tracking Multiple Moving Object Using Gaussian Mixture Model". **International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-2**

- [14] J.P Lewis.(1995). "Fast normalized Cross Correlation". **Industrial Light and Magic.**

- [15] Briechle, Kai dan Hanebeck, Uwe D. (2001). "Template Matching Using Fast Normalized Cross Correlation". **Institute of Automatic Control Engineering, Technische Universitat Muenchen.**
- [16] Takita, Kenji et all. (2003). "High-accuracy subpixel image registration based on phase-only correlation". **IECE Trans. Fundamentals. Vol.E86-A**

“Halaman ini sengaja dikosongkan”

DAFTAR LAMPIRAN

	Halaman
Lampiran A : Kode Program Proses Antarmuka Awal.....	71
Lampiran B : Kode Program Normalized Cross Correlation..	75
Lampiran C : Kode Program PBIM.....	83
Lampiran D : Kode Program POCS	89

“Halaman ini sengaja dikosongkan”

LAMPIRAN A

Kode Program Proses Antarmuka Awal

```

function varargout = Resolusi(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',           mfilename, ...
...
'gui_Singleton',  gui_Singleton, ...
'gui_OpeningFcn', @Resolusi_OpeningFcn, ...
'gui_OutputFcn',  @Resolusi_OutputFcn, ...
'gui_LayoutFcn',  [] , ...
'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = ...
gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function Resolusi_OpeningFcn(hObject, eventdata,
handles, varargin)
handles.output = hObject;
ha = axes('units','normalized', ...
'position',[0 0 1 1]);
% Move the background axes to the bottom
uistack(ha,'bottom');
I=imread('gambar2.jpg');
hi = imagesc(I);
set(ha,'handlevisibility','off', ...
'visible','off')
guidata(hObject, handles);
function varargout = Resolusi_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

```

LAMPIRAN A (LANJUTAN)

```

function Home_Callback(hObject, eventdata,
handles)
Home
function file_Callback(hObject, eventdata,
handles)
function pilih_file_Callback(hObject, eventdata,
handles)
proyek=guidata(gcbo);
global vid img
[filename, pathname] = uigetfile({'*avi'},
'Pilih Video'); % akses directory file video
vid=strcat(pathname, filename);
videoReader = vision.VideoFileReader(vid,
'ImageColorSpace','Intensity')
img = step(videoReader);
axes(handles.axes1);
imshow(img);
function reset_Callback(hObject, eventdata,
handles)
function roi_Callback(hObject, eventdata,
handles)% Executes on button press in roi.
proyek=guidata(gcbo);
global vid img get_roi target_img
h = imrect;
posisi = wait(h);
get_roi = getPosition(h)
centroid = [get_roi(1)+uint32(get_roi(3)/2)
get_roi(2)+uint32(get_roi(4)/2)]
target_img = imcrop(img,get_roi);
axes(handles.axes4);
imshow(target_img);
function play_Callback(hObject, eventdata,
handles)
ncc

function sampel_Callback(hObject, eventdata,
handles)

```

LAMPIRAN A (LANJUTAN)

```

global im s factor
factor = 2;
s=[];
s=num2cell(s);
im=num2cell(im);
n = get(handles.jml, 'String');
n = str2num (n);
for i = 1:n
    im{i} = imread(['D:\Habib TA-112\MATLAB
SR\Matlab(2)\image' num2str(i) '.jpg']);
    s{i} = imread(['D:\Habib TA-112\MATLAB
SR\Matlab(2)\image' num2str(i) '.jpg']);
end
msgbox('Selesai,proses input data sukses')
function processing_Callback(hObject, eventdata,
handles)
function reg_Callback(hObject, eventdata,
handles)
global im delta_est t1
tic
[delta_est, phi_est] = PBIM(im)
msgbox('Proses Registrasi Selesai')
toc
t1=toc;
function rec_Callback(hObject, eventdata,
handles)
global s delta_est factor t1 target_img
tic
y = pocs(s,delta_est,factor)
imwrite(y,'D:\\Habib TA-112\Citra Super
Resolusi.jpg');
axes(handles.axes5);
imshow(y);
toc
t2=toc;
time=t1+t2
set(handles.time, 'String',time);

```

LAMPIRAN A (LANJUTAN)

```

function jml_Callback(hObject, eventdata,
handles)
function jml_CreateFcn(hObject, eventdata,
handles)
if ispc &&
isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function Uji_SR_Callback(hObject, eventdata,
handles)
Pengujian_Super_Resolusi
function time_Callback(hObject, eventdata,
handles)
function time_CreateFcn(hObject, eventdata,
handles)
if ispc &&
isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```

LAMPIRAN B

Kode Program Normalized Cross Correlation

```

global vid
threshold = single(0.99);
level = 2;
pathDir = 'D:\Habib TA-112\MATLAB SR\Matlab(2)';
workingDir = 'D:\Habib TA-112\MATLAB
SR\Matlab(2)';

% membaca video
hVideoSrc = vision.VideoFileReader(vid, ...
'VideoOutputDataType', 'single', ...
'ImageColorSpace', 'Intensity');

% gaussian piramid sistem objek digunakan untuk
dekomposisi template dan
% objek dalam frame, sehingga nilai cross
correlation dapat dihitung dalam
% area objek, bukan keseluruhan frame
hGaussPymd1 =
vision.Pyramid('PyramidLevel',level);
hGaussPymd2 =
vision.Pyramid('PyramidLevel',level);
hGaussPymd3 =
vision.Pyramid('PyramidLevel',level);

% rotasi citra dengan sudut pi sebelum komputasi
perkalian dengan target
% di domain frekuensi yang ekuivalen dengan
correlation
hRotatel = vision.GeometricRotator('Angle', pi);

hFFT2D1 = vision.FFT; % untuk target

```


LAMPIRAN B (LANJUTAN)

```

hFFT2D2 = vision.FFT; % untuk citra testing

% sistem objek untuk invers FFT 2D setelah
correlation di domain frekuensi.
hIFFFT2D = vision.IFFT;

% sistem objek konvolusi 2dimensi untuk merata-
rata kekuatan citra dlm
% rentangan dimensi target yang sama
hConv2D =
vision.Convolver('OutputSize','Valid');

useDefaultTarget = true; % menentukan jumlah
target yang akan di tracking
[Img, numberOfTargets, target_image] = ... %
fungsi untuk melakukan tracking
proses_ncc_temp(useDefaultTarget);

% Downsample target dg gaussian pyramid
u/mengurangi jumlah komputasi cross
% correlation
target_image = single(target_image);
target_dim_nopyramid = size(target_image);
target_image_gp = step(hGaussPymdl,
target_image);
target_energy =
sqrt(sum(target_image_gp(:).^2));

target_image_rot = step(hRotatel,
target_image_gp); % merotasi target 180'

```

LAMPIRAN B (LANJUTAN)

```

[rt, ct] = size(target_image_rot); %
mengembalikan ukuran matrix ke variabel rt dan
ct scr terpisah
Img = single(Img); % konversi ke ketelitian
tunggal
Img = step(hGaussPyemd2, Img);
[ri, ci]= size(Img);% mengembalikan ukuran
matrix Img ke variabel ri dan ci scr terpisah
r_mod = 2^nextpow2(rt + ri);
c_mod = 2^nextpow2(ct + ci);
target_image_p = [target_image_rot zeros(rt,
c_mod-ct)]; % penyamaan dimensi
target_image_p = [target_image_p; zeros(r_mod-
rt, c_mod)];

target_fft = step(hFFT2D1, target_image_p); % 2D
FFT pada citra target

% inisialisasi variabel konstanta untuk proses
iterasi
target_size = repmat(target_dim_nopyramid,
[numberOfTargets, 1]);
gain = 2^(level);
Im_p = zeros(r_mod, c_mod, 'single'); %
digunakan untuk zero padding
C_ones = ones(rt, ct, 'single'); %
digunakan untuk menghitung rata2 dg konvolusi

% sistem objek untuk menghitung nilai local
maximum dari normalized cross correlation.
hFindMax = vision.LocalMaximaFinder( ...
'Threshold', single(-1), ...
'MaximumNumLocalMaxima', numberOfTargets, ...
'NeighborhoodSize',
floor(size(target_image_gp)/2)*2 - 1);

```

LAMPIRAN B (LANJUTAN)

```
% sistem objek untuk menampilkan tracking objek
sz = get(0, 'ScreenSize');
pos = [20 sz(4)-400 400 300];
hROIIPattern = vision.VideoPlayer('Name',
'Tracking Object', ...
'Position', pos);

% plotting normalized cross correlation
hPlot =
videopatternplots('setup', numberOfTargets,
threshold);
i = i;
% Proses Looping pengolahan video
while ~isDone(hVideoSrc)
    Im = step(hVideoSrc);
    Im_gp = step(hGaussPyemd3, Im);

% konvolusi domain frekuensi.
    Im_p(1:ri, 1:ci) = Im_gp;    % Zero-pad
    img_fft = step(hFFT2D2, Im_p);
    corr_freq = img_fft .* target_fft;
    corrOutput_f = step(hIFFFT2D, corr_freq);
    corrOutput_f = corrOutput_f(rt:ri, ct:ci);

% menghitung kekuatan citra dan blok berjalan
sesuai ukuran target
% template
    IUT_energy = (Im_gp).^2;
    IUT = step(hConv2D, IUT_energy, C_ones);
    IUT = sqrt(IUT);

% menghitung normalized cross correlation.
```

LAMPIRAN B (LANJUTAN)

```

        norm_Corr_f = (corrOutput_f) ./ (IUT *
target_energy);
        xyLocation = step(hFindMax, norm_Corr_f);

% menghitung indeks linier
        linear_index = sub2ind([ri-rt, ci-ct]+1,
xyLocation(:,2),...
        xyLocation(:,1));

        norm_Corr_f_linear = norm_Corr_f(:);
        norm_Corr_value =
norm_Corr_f_linear(linear_index);
        detect = (norm_Corr_value > threshold);
        target_roi = zeros(length(detect), 4);
        ul_corner = (gain.*(xyLocation(detect, :)-
1))+1;
        target_roi(detect, :) = [ul_corner,
fliplr(target_size(detect, :))];

% menggambar bounding box.
        Imf = insertShape(Im, 'Rectangle',
target_roi, 'Color', 'green');
        img_cropp = imcrop(Im ,target_roi);
% memberi label citra
        filename=[sprintf('image%d',i) '.jpg'];
        fullname= fullfile(workingDir,filename);
        imwrite(img_cropp,fullname);
% Plot normalized cross correlation.
        plot_ncc('update',hPlot,norm_Corr_value);
        step(hROIPattern, Imf);
        i = i+1;

end

release(hVideoSrc);

```

LAMPIRAN B (LANJUTAN)

```
%fungsi untuk processing template
function [img, numTargets, target_img] =
proses_ncc_temp(useDefaultTarget)
%[img, numTargets, target_img] = nilai balik
%useDefaultTarget = parameter

global vid get_roi fcn
% pathDir = 'D:\Habib TA-112\MATLAB
SR\Matlab(2)';
% workingDir = 'D:\Habib TA-112\MATLAB
SR\Matlab(2)';
numTargets = 1;

% membaca frame pertama
reader = vision.VideoFileReader(vid, ...
'VideoOutputDataType', 'single',...
'ImageColorSpace', 'Intensity');
img = step(reader);
release(reader);

% menentukan lokasi template
roi = get_roi;
target_img = imcrop(img,roi);
% filename=[sprintf('image0.jpg')];
% fullname= fullfile(workingDir,filename);
% imwrite(target_img,fullname);

hf = figure('Color', get(0,
'defaultuicontrolbackgroundcolor'), ...
'Name', 'Target pattern', ...
'NumberTitle', 'off');
imshow(img);

if useDefaultTarget
    numTargets = 1;
% menampilkan template
```

LAMPIRAN B (LANJUTAN)

```

        rectangle('Position', roi, 'EdgeColor',[0 1
0]);
        roi_rect=rectangle;
        pause(1);
        close(hf);
return;
else

        h = imrect(gca, roi);
        api = iptgetapi(h);
        api.setColor([0 1 0]);
        api.addNewPositionCallback(@(p)
title(mat2str(p)));

        fcn =
makeConstrainToRectFcn('imrect',get(gca,'XLim'),
get(gca,'YLim'));
        api.setDragConstraintFcn(fcn);

        yshift = 10;
        uicontrol(hf, 'style', 'text', 'Units',
'Pixels', ...
'String', 'Number of targets:', ...
'FontSize', 12, ...
'Position', [80 yshift 150 20]);
        hEditBox = uicontrol(hf, 'style', 'edit',
'Units', 'Pixels', ...
'String', '1', ...
'HorizontalAlignment', 'left', ...
'BackgroundColor', [1 1 1], ...
'Position', [230 yshift 100 20]);
        uicontrol(hf, 'style', 'pushbutton',
'Units', 'Pixels', ...
'String', 'Submit', ...
'Position', [340 yshift 100 20], ...
'Callback', @submitFcn);

```

LAMPIRAN B (LANJUTAN)

```

        uiwait;
end

function submitFcn(varargin)
    roi = api.getPosition();

    % ekstrak data template
    target_img = imcrop(img,roi);
    % ('base','target_img', target_img);
    if any(size(target_img) < 20) ||
    any(size(target_img) > 100)
        errordlg('Target height and width
must be between 20 and 100 pixels.',...
'Invalid dimensions');
    return;
end

    numTargets =
round(str2double(get(hEditBox, 'String')));
    if numTargets < 1
        warndlg('Number of targets must be
greater than or equal to 1. Setting the number
of targets to 1.', 'Invalid number of targets');
        numTargets = 1;
    end

    close(hf);
end
end

```

LAMPIRAN C

Kode Program PBIM

```

function [delta_est, phi_est] = PBIM(im)
for imnr = 2:length(im)
    im0{1} = im{1};
    im1{1} = im{imnr};
    [output shift] =
dftregistration(fft2(im0{1}),fft2(im1{1}),1000);
    phi_est(imnr) = output(1,2);
    delta_est(imnr,:) = output(3:4);
end

function [output Greg] =
dftregistration(buf1ft,buf2ft,usfac)
if usfac == 0,
    CCmax = sum(sum(buf1ft.*conj(buf2ft)));
    rfzero = sum(abs(buf1ft(:)).^2);
    rgzero = sum(abs(buf2ft(:)).^2);
    error = 1.0 -
CCmax.*conj(CCmax)/(rgzero*rfzero);
    error = sqrt(abs(error));
    diffphase=atan2(imag(CCmax),real(CCmax));
    output=[error,diffphase];

elseif usfac == 1,
    [m,n]=size(buf1ft);
    CC = ifft2(buf1ft.*conj(buf2ft));
    [max1,loc1] = max(CC);
    [max2,loc2] = max(max1);
    rloc=loc1(loc2);
    cloc=loc2;
    CCmax=CC(rloc,cloc);
    rfzero = sum(abs(buf1ft(:)).^2)/(m*n);
    rgzero = sum(abs(buf2ft(:)).^2)/(m*n);
    error = 1.0 -
CCmax.*conj(CCmax)/(rgzero(1,1)*rfzero(1,1));
    error = sqrt(abs(error));

```


LAMPIRAN C (LANJUTAN)

```

        diffphase=atan2(imag(CCmax),real(CCmax));
        md2 = fix(m/2);
        nd2 = fix(n/2);
    if rloc > md2
        row_shift = rloc - m - 1;
    else
        row_shift = rloc - 1;
    end

    if cloc > nd2
        col_shift = cloc - n - 1;
    else
        col_shift = cloc - 1;
    end

    output=[error,diffphase,row_shift,col_shift];

    % Partial-pixel shift
    else
    % First upsample by a factor of 2 to obtain
    initial estimate
    % Embed Fourier data in a 2x larger array
        [m,n]=size(buf1ft);
        mlarge=m*2;
        nlarge=n*2;
        CC=zeros(mlarge,nlarge);
        CC(m+1-fix(m/2):m+1+fix((m-1)/2),n+1-
        fix(n/2):n+1+fix((n-1)/2)) = ...

    fftshift(buf1ft).*conj(fftshift(buf2ft));

    % Compute crosscorrelation and locate the peak
        CC = ifft2(ifftshift(CC)); % Calculate
    cross-correlation
        [max1,loc1] = max(CC);
        [max2,loc2] = max(max1);

```

LAMPIRAN C (LANJUTAN)

```

rloc=loc1(loc2);cloc=loc2;
CCmax=CC(rloc,cloc);

% Obtain shift in original pixel grid from the
% position of the
% crosscorrelation peak
[m,n] = size(CC); md2 = fix(m/2); nd2 =
fix(n/2);
if rloc > md2
    row_shift = rloc - m - 1;
else
    row_shift = rloc - 1;
end
if cloc > nd2
    col_shift = cloc - n - 1;
else
    col_shift = cloc - 1;
end
row_shift=row_shift/2;
col_shift=col_shift/2;
% menengahkan posisi peak

% If upsampling > 2, then refine estimate with
matrix multiply DFT
if usfac > 2,
    %% DFT computation %%
    % Initial shift estimate in upsampled grid

    row_shift =
round(row_shift*usfac)/usfac;
    col_shift =
round(col_shift*usfac)/usfac;
    dftshift = fix(ceil(usfac*1.5)/2); %%
Center of output array at dftshift+1
% Matrix multiply DFT around the current shift
estimate

```

LAMPIRAN C (LANJUTAN)

```

        CC =
conj(dftups(buf2ft.*conj(buf1ft),ceil(usfac*1.5)
,ceil(usfac*1.5),usfac,...
        dftshift-row_shift*usfac,dftshift-
col_shift*usfac))/(md2*nd2*usfac^2);
% Locate maximum and map back to original pixel
grid
        [max1,loc1] = max(CC);
        [max2,loc2] = max(max1);
        rloc = loc1(loc2); cloc = loc2;
        CCmax = CC(rloc,cloc);
        rg00 =
dftups(buf1ft.*conj(buf1ft),1,1,usfac)/(md2*nd2*
usfac^2);
        rf00 =
dftups(buf2ft.*conj(buf2ft),1,1,usfac)/(md2*nd2*
usfac^2);
        rloc = rloc - dftshift - 1;
        cloc = cloc - dftshift - 1;
        row_shift = row_shift + rloc/usfac;
        col_shift = col_shift + cloc/usfac;

% If upsampling = 2, no additional pixel shift
refinement
else
        rg00 = sum(sum( buf1ft.*conj(buf1ft)
))/m/n;
        rf00 = sum(sum( buf2ft.*conj(buf2ft)
))/m/n;
end
        error = 1.0 -
CCmax.*conj(CCmax)/(rg00*rf00);
        error = sqrt(abs(error));
        diffphase=atan2(imag(CCmax),real(CCmax));
% If its only one row or column the shift along
that dimension has no
% effect. We set to zero.

```

LAMPIRAN C (LANJUTAN)

```

if md2 == 1,
    row_shift = 0;
end
if nd2 == 1,
    col_shift = 0;
end

output=[error,diffphase,row_shift,col_shift];
end

% Compute registered version of buf2ft
if (nargout > 1)&&(usfac > 0),
    [nr,nc]=size(buf2ft);
    Nr = ifftshift([-fix(nr/2):ceil(nr/2)-1]);
    Nc = ifftshift([-fix(nc/2):ceil(nc/2)-1]);
    [Nc,Nr] = meshgrid(Nc,Nr);
    Greg = buf2ft.*exp(i*2*pi*(-row_shift*Nr/nr-
col_shift*Nc/nc));
    Greg = Greg*exp(i*diffphase);
elseif (nargout > 1)&&(usfac == 0)
    Greg = buf2ft*exp(i*diffphase);
end
return

function out=dftups(in,nor,noc,usfac,roff,coff)

[nr,nc]=size(in);
% Set defaults
if exist('roff')~=1, roff=0; end
if exist('coff')~=1, coff=0; end
if exist('usfac')~=1, usfac=1; end
if exist('noc')~=1, noc=nc; end
if exist('nor')~=1, nor=nr; end
% Compute kernels and obtain DFT by matrix
products

```

LAMPIRAN C (LANJUTAN)

```

kernc=exp((-i*2*pi/(nc*usfac))*(
ifftshift([0:nc-1]).' - floor(nc/2) )*( [0:noc-
1] - coff ));
kernr=exp((-i*2*pi/(nr*usfac))*( [0:nor-1].' -
roff )*( ifftshift([0:nr-1]) - floor(nr/2) ));
out=kernr*in*kernc;
return

```

LAMPIRAN D

Kode Program POCS

```
function y = pocs(s,delta_est,factor)

max_iter = 50;

temp = upsample(upsample(s{1}, factor)',
factor)';
y = zeros(size(temp));
coord = find(temp);
y(coord) = temp(coord);

for i = 2:length(s)
    temp = upsample(upsample(s{i}, factor)',
factor)';
    temp = shift(temp, round(delta_est(i,
2)*factor), round(delta_est(i, 1)*factor));
    coord = find(temp);
    y(coord) = temp(coord);
end

y_prev=y;

E=[];
iter=1;

blur = [.25 0 1 0 .25;...
        0 1 2 1 0;...
        1 2 4 2 1;...
        0 1 2 1 0;...
        .25 0 1 0 .25];

blur = blur / sum(blur(:));
```

LAMPIRAN D (LANJUTAN)

```

wait_handle = waitbar(0, 'Reconstruction...',
'Name', 'SuperResolution GUI'); %not required
rif...!!! :-)

while iter < max_iter
    waitbar(min(4*iter/max_iter, 1),
wait_handle); %not required rif...!!! :-)
    y = imfilter(y, blur);
for i = length(s):-1:1
    temp = upsample(upsample(s{i}, factor)',
factor)';
    temp = shift(temp, round(delta_est(i,
2)*factor), round(delta_est(i, 1)*factor));
    coord = find(temp);
    y(coord) = temp(coord);
end

    delta= norm(y-y_prev)/norm(y);
    E=[E; iter delta];
    iter = iter+1;
if iter>3
if abs(E(iter-3,2)-delta) <1e-4
break
end
end
    y_prev=y;
end
y = uint8(y);

y(1,:)

close(wait_handle);

```

BIODATA PENULIS



Penulis memiliki nama lengkap Hasanuddin Al-Habib, lahir di Mojokerto pada tanggal 9 Agustus 1993. Penulis berasal dari Kota Mojokerto, bertempat tinggal di Ds. Jambuwok RT.03/RW.02 Kec. Trowulan, Kab. Mojokerto. Pendidikan formal yang pernah ditempuh yaitu MI AL-HUDA Jambuwok, SMP Negeri 2 Trowulan, dan MAN Mojokerto. Kemudian, penulis melanjutkan studi di jurusan Matematika ITS, dengan bidang minat ilmu

komputer. Dalam bidang minat ini penulis mulai mengenal bahasa pemrograman diantaranya adalah C, C++, Java, PHP-MySQL, dan MATLAB. Semasa menempuh jenjang pendidikan S-1, penulis juga aktif dalam kegiatan non-akademis diantaranya aktif di organisasi kemahasiswaan Matematika ITS dan mengikuti kepanitiaan acara besar yang ada di ITS diantaranya: OMITS, dan KNM-17. Selama penulisan Tugas Akhir ini Penulis tidak lepas dari kekurangan, untuk itu penulis mengharapkan kritik, saran, dan pertanyaan mengenai Tugas Akhir ini yang dapat dikirimkan melalui *e-mail* ke hasanuddin.alhabib@gmail.com.